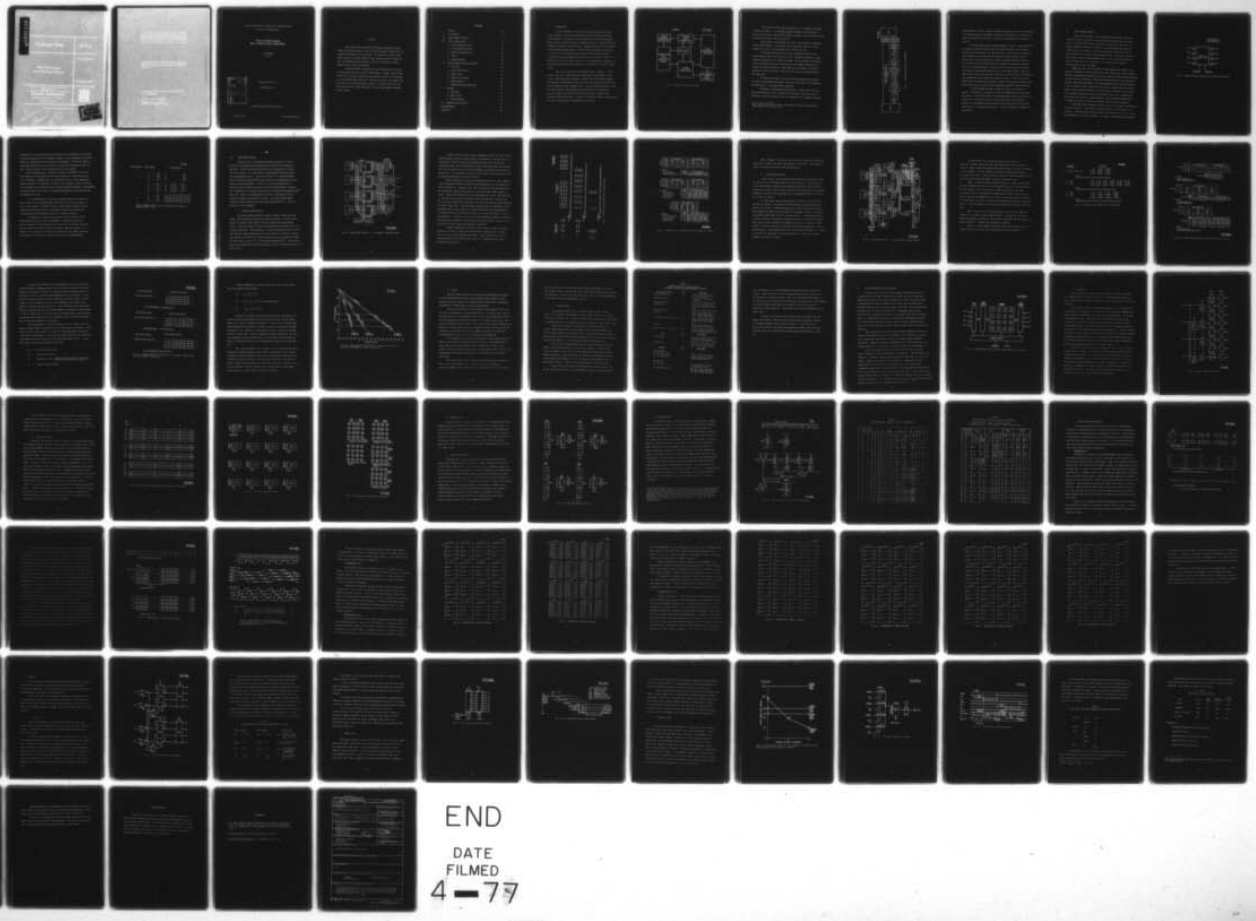
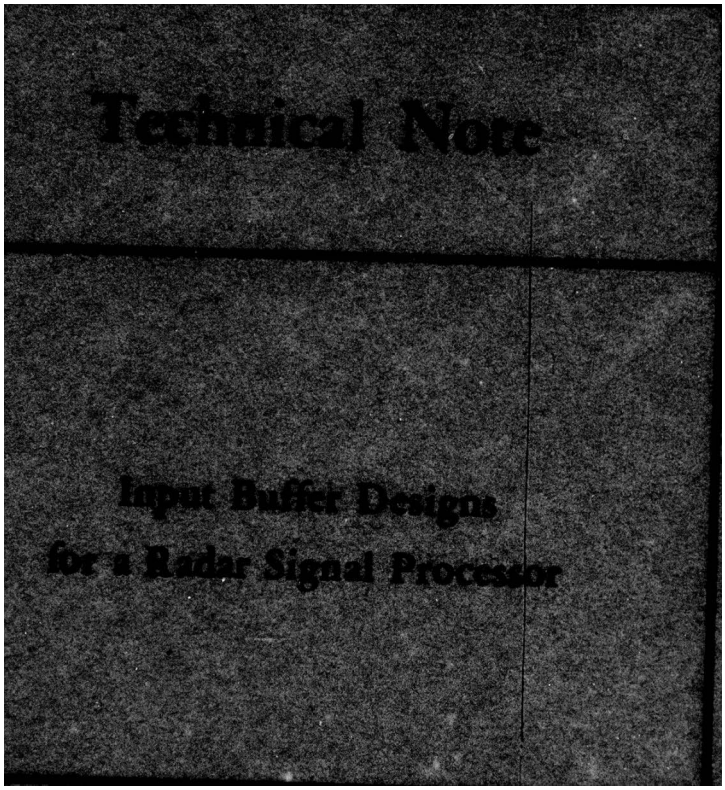
AD-A037 120 MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB F/G 9/2
INPUT BUFFER DESIGNS FOR A RADAR SIGNAL PROCESSOR.(U)
JAN 77 A H HUNTOON F19628-76-C-0002
UNCLASSIFIED TN-1977-3 ESD-TR-77-26 NL

1 OF 1
AD A037 120



END

DATE FILMED
4-77


Technical Note
Input Buffer Designs
for a Radar Signal Processor

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

LINCOLN LABORATORY

INPUT BUFFER DESIGNS FOR A RADAR SIGNAL PROCESSOR

A. H. HUNTOON

Group 27

TECHNICAL NOTE 1977-3

20 JANUARY 1977

LEXINGTON MASSACHUSETTS

Abstract

Three Input Buffer designs which provide intermediate storage between the Analog-to-Digital Converters and the Digital Matched Filter of a Radar Signal Processor are presented. All designs use a basic buffer module having a selectable number of input channels and data storage formats, and differing from one another in terms of the number of input channels and/or levels of buffering.

Prototype hardware experiments have been conducted on the speed-constraining portions of an all-ECL buffer module. Results indicate that the use of 8:1 input data demultiplexing together with ECL 10K technology will yield buffer input rates up to 142 Ms/s per channel, and output rates of 45 Ms/s per rail to the digital convolver. The discussion also considers the use of ECL 100K technology to achieve input speeds of approximately 190 Ms/s.

CONTENTS

I. INTRODUCTION

Lincoln Laboratory is presently developing an advanced digital signal processor for the Ballistic Missile Defense Advanced Technology Center (BMDATC). The objective of the program is to develop high-speed, highly-flexible digital processing techniques and then to demonstrate these techniques through the fabrication of a signal processor to be incorporated in the System Technology Radar. The Input Buffer (hereafter referred to as the IB) is one subsystem of the processor and this report presents initial designs and experimental results for this subsystem. The IB is a subsystem of the radar signal processor. It is appropriate, therefore, to review the signal processor structure in the context of the overall radar system.

Most radar systems have the structure shown in Figure 1. These systems consist of a Data Processing System (DPS) which provides overall system control and decision making, a control unit which decodes commands and provides the appropriate timing, the array for generating, transmitting and receiving RF energy together with its exciter and beam controller, and the signal processor which generates the waveforms to be transmitted, provides matched filtering for the received waveforms, generates the necessary target metrics and, in general, selectively reduces the data volume and rate to values which are appropriate for the DPS.

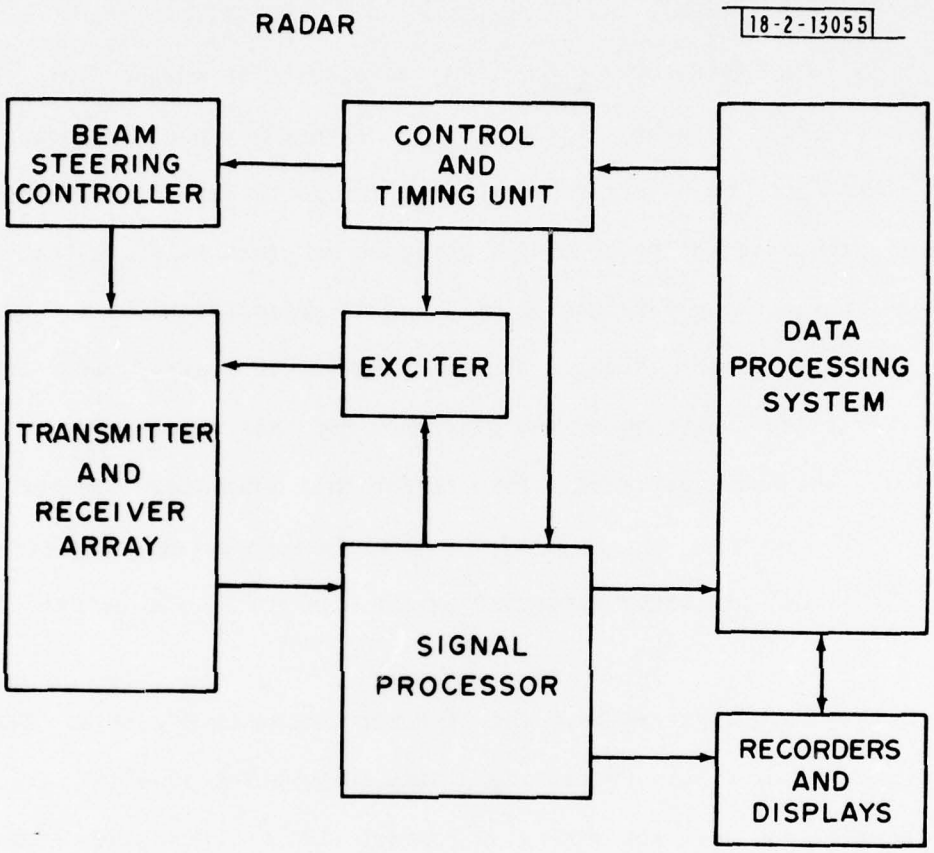


Fig. 1. Radar system block diagram.

The structure of the signal processor as it is presently envisioned is shown in Figure 2. The analog-to-digital converters will be pairs of 10-bit 60 Ms/s devices which deliver parallel channels of 22 bit words (10 bits plus parity for both I and Q) to the IB.*

Three designs are described for the IB. All designs are based upon a basic buffer module. In each case, the IB input data rate is at least 60 Ms/s per channel, and the output rate is 30 Ms/s per rail.

The Digital Convolver System provides matched filtering of the received radar signal stored in the IB. This subsystem is based on a radix 4, 16384-point pipelined FFT, clocked at 30 MHz. The DCS performs filtering by Fourier transforming the data, multiplying in the frequency domain by the selected reference function, and then using the same hardware switched to perform the inverse transform. The output is thus the filtered version of the input data.

The Detection Processor follows the DCS and compares the filtered data with a predetermined threshold. Both data and results of the comparison are then passed to the Output Buffer subsystem.

Subsequent to filtering and thresholding, a traditional set of post processing operations, such as monopulse calculations, range and velocity estimation, bulk filtering and others is required prior to further processing

---

* Input Buffer word lengths will be later shown as 24 bits, which permits the eventual use of 11-bit A/D's.

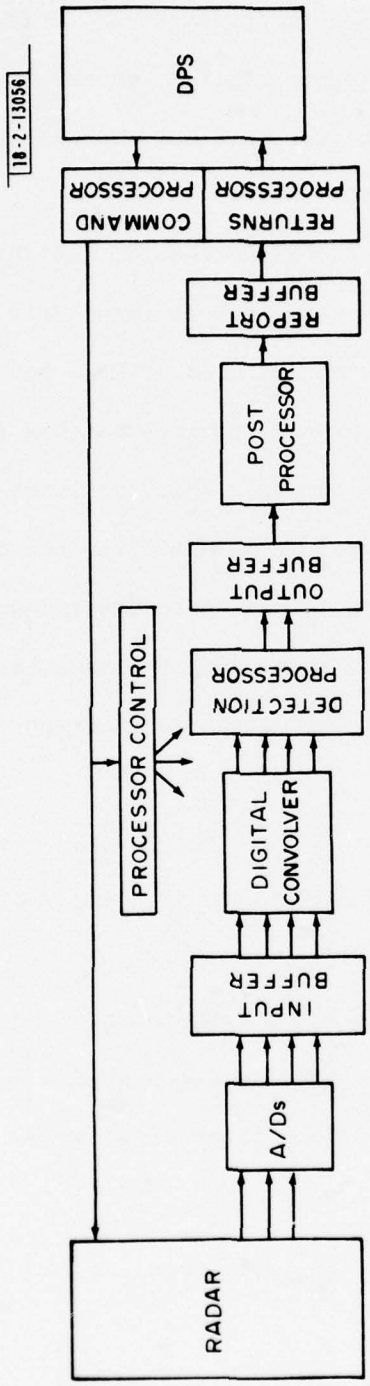


Fig. 2. Signal Processor system diagram.

within the DPS. The Post Processor performs these functions, and is envisioned as a high-performance programmable machine whose architecture is tailored for radar post processing algorithms. Alteration of algorithms may be done in software.

The three buffer memory subsystems shown in Figure 2 allow computation to proceed at differing rates within the Digital Convolver, Post Processor and Data Processing System. An important point to note is that the amount of time consumed at each stage must not exceed the radar interpulse period. While the IB collects raw data for eventual processing within the convolver, the convolver may be operating upon IB data already in storage, passing the results to the Output Buffer. Similarly, the Post Processor can be taking data from the Output Buffer and placing it in the Report Buffer. Thus, the three major stages of the processor (the A/D's, the DCS and the Post Processor) may be operating on individual and separate data sets. The IB is the first buffer in this pipeline flow and must be capable of smoothing the entire radar data stream so that it can be evenly processed through the signal processor.

In the following sections, designs for the IB are presented in detail. In Section II, an overview of the basic buffer module is presented. This module is used in the three system configurations which are discussed in Section III. The detailed design of the memory module is described in Section IV, and hardware prototype efforts, which establish the anticipated speed limits, are discussed in Section V. Section VI presents a summary and conclusions.

II. BUFFER MODULE OVERVIEW

Because the IB serves as the intermediary between the 60 Ms/s A/D converters and the 30 Ms/s DCS, it must provide input and output data rates equal to or greater than 60 Ms/s and 30 Ms/s, respectively. The size or storage capacity of the IB is a function of the radar operating modes and transform sizes anticipated. In the signal processor, the matched filter (DCS) performs 16K, 8K and 4K point convolution. This limits the time bandwidth product of the incoming baseband signal to less than 16K and places a bound on minimum IB size at 16K words. As a result, the IB design is centered around a basic 16K module.

Though the final choice on number of input channels may differ, let us assume for purposes of this discussion that the IB may have up to four channels. Moreover, the designs presented here consider double-or triple-buffered operation for 16K, 8K and 4K transforms. Beyond the buffering function, the IB must also reformat stored data and present it in the appropriate order for the convolver. A final distinction must be made between the non-overlapped, multiple-buffered modes in which single data batches are processed and a separate class of modes for all-range processing in which stored batches of data are overlapped when presented to the convolver. When operating in the overlapped configuration, it should be possible for the IB to accept and deliver data continuously.

A simplified block diagram of the basic 16K x 24 half-duplex buffer module is shown in Figure 3. This structure always provides a 4-rail output to the convolver. The design presented here permits the selection of 1, 2, or 4 active input channels to the basic 16K module. In Figure 3 and subsequently throughout

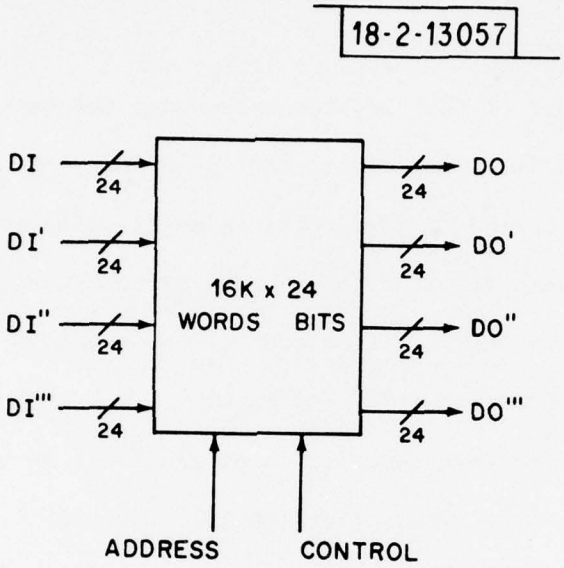


Fig. 3. Simplified block diagram, half duplex buffer module.

this report, data are labelled in accordance with the associated radar channel of origin. Unprimed data points originate on channel 1, singly primed data is associated with channel 2, etc.

The remainder of this section introduces the basic 16K x 24 buffer module in terms of its 9 selectable configurations. The manner in which these modules may be assembled to form complete multi-buffered systems capable of providing output data in either overlapped or non-overlapped format is discussed in Section III. The ability to select 1, 2 or 4 active input channels per module makes it generally useful for building up larger buffer memory systems.

A detailed hardware description of the 16K x 24 buffer module will be presented later in this report (Section IV). However, let us first characterize module operation in terms of the number of active input channels (1, 2 or 4) together with the necessary data formatting for performing 4K, 8K or 16K transforms within the DCS. Nine canonic module configurations are described in Figures 4 and 5 in terms of input and output data streams. The use of individual modules in their two or four input channel configurations is of interest when either the utmost input speed is not a requirement, or when the number of modules per system is limited. In the buffer memory designs to be later described, all modules are employed in their high speed, single input channel configuration.

The first three module configurations apply to single input channel operation and provide correct output data formatting for a single 16K transform, two 8K transforms, or four 4K transforms, respectively.

Configurations 4 through 6 of Figure 4 are for two-channel operation. Configuration 4 describes the collection of two 8K-point data sets on separate channels, and subsequent memory readout in "packed" format for 16K transform processing.

CONFIGURATION	INPUT STREAM	OUTPUT STREAM
1	0 → 16K-1	0 → 4K-1
		4K → 8K-1
		8K → 12K-1
		12K → 16K-1
2	0 → 16K-1	0 → 2K-1, 8K → 10K-1
		2K → 4K-1, 10K → 12K-1
		4K → 6K-1, 12K → 14K-1
		6K → 8K-1, 14K → 16K-1
3	0 → 16K-1	0 → 1K-1, 4K → 5K-1, 8K → 9K-1, 12K → 13K-1
		1K → 2K-1, 5K → 6K-1, 9K → 10K-1, 13K → 14K-1
		2K → 3K-1, 6K → 7K-1, 10K → 11K-1, 14K → 15K-1
		3K → 4K-1, 7K → 8K-1, 11K → 12K-1, 15K → 16K-1
4	0 → 8K-1	0 → 4K-1
	0′ → 8K-1′	4K → 8K-1
		0′ → 4K-1′
		4K′ → 8K-1′
5	0 → 8K-1	0 → 2K-1, 0′ → 2K-1′
	0′ → 8K-1′	2K → 4K-1, 2K′ → 4K-1′
		4K → 6K-1, 4K′ → 6K-1′
		6K → 8K-1, 6K′ → 8K-1′
6	0 → 8K-1	0 → 1K-1, 4K → 5K-1, 0′ → 1K-1′, 4K′ → 5K-1′
	0′ → 8K-1′	1K → 2K-1, 5K → 6K-1, 1K′ → 2K-1′, 5K′ → 6K-1′
		2K → 3K-1, 6K → 7K-1, 2K′ → 3K-1′, 6K′ → 7K-1′
		3K → 4K-1, 7K → 8K-1, 3K′ → 4K-1′, 7K′ → 8K-1′

18-2-13196

Fig. 4. Memory module input and output data streams for Configurations 1 to 6.

Configuration 5 is arranged such that successive 8K transforms may be performed on data associated with two independent channels, while configuration 6 provides proper data ordering for four successive 4K transforms. The 8K data samples on each input channel need not necessarily be viewed as 8K contiguous points, but could instead actually be two concatenated 4K sets per channel.

Module configurations 7 through 9 of Figure 5 accommodate four-channel data inputs, and again provide the necessary outputs for calculating 16K, 8K or 4K transforms. Configuration 7 has additional utility in that output data remains in exactly the same order in which it was collected, and may, therefore, be reintroduced into the buffer input for either playback operation or reformatting. In playback modes the IB data is recorded and played back in non-real time for processing by the DCS.

It is approximately true that the maximum sample rate at which data may be stored in an individual buffer module does not vary as the number of active input channels is altered. That is, single channel usage of a given buffer module permits data rates (per channel) which are twice those for 2-channel operation and quadruple those for 4-channel operation.

The operation of several memory modules within a multi-buffered memory system implies the use of independent, asynchronous controllers. The discussion to follow presents three representative IB designs, differing from one another according to depth of buffering, number of channels, and whether or not output data sets may be overlapped for "all-range" processing. Consideration is also given to questions of parity and command formats.

18-2-13195

CONFIGURATION	INPUT STREAM			OUTPUT STREAM				
7	0	→	4K-1	0	→			4K-1
	0′	→	4K-1′	0′	→			4K-1′
	0″	→	4K-1″	0″	→			4K-1″
	0‴	→	4K-1‴	0‴	→			4K-1‴
8	0	→	4K-1	0	→	2K-1, 0″	→	2K-1″
	0′	→	4K-1′	2K	→	4K-1, 2K″	→	4K-1″
	0″	→	4K-1″	0′	→	2K-1′, 0‴	→	2K-1‴
	0‴	→	4K-1‴	2K′	→	4K-1′, 2K‴	→	4K-1‴
9	0	→	4K-1	0 → 1K-1, 0′ → 1K-1′, 0″ → 1K-1″, 0‴ → 1K-1‴				
	0′	→	4K-1′	1K → 2K-1, 1K′ → 2K-1′, 1K″ → 2K-1″, 1K‴ → 2K-1‴				
	0″	→	4K-1″	2K → 3K-1, 2K′ → 3K-1′, 2K″ → 3K-1″, 2K‴ → 3K-1‴				
	0‴	→	4K-1‴	3K → 4K-1, 3K′ → 4K-1′, 3K″ → 4K-1″, 3K‴ → 4K-1‴				

Fig. 5. Memory module input and output data streams for Configurations 7 to 9.

III. INPUT BUFFER DESIGNS

The basic 16K x 24 reconfigurable module of Section II may be employed to construct multi-channeled, multi-leveled buffer memories in many ways. The discussion below describes three representative systems. The first, System 1, is a four-channel, double-buffered structure having non-overlapped data access. The second, System 2, is a three-channeled buffer featuring triple-buffering and the ability to simultaneously accommodate an auxiliary channel. System 3 is a special configuration of System 2 which illustrates some ways in which a flexible memory structure may be employed to store data continuously while delivering overlapped output data sets. Finally, consideration will be given to questions of parity checking and memory system control formats. All system designs employ single-rail module inputs, hence only module configurations 1, 2 and 3 of Section II will be employed.

A. Double-Buffered System

The double-buffered structure shown in Figure 6 allows full four-channel input capability together with high input speed. All eight buffer modules use single-channel inputs which will support data rates beyond 60 Ms/s. They are connected to four high-performance A/D converter pairs via an input multiplexer. Correspondingly, an output multiplexer collects 4-rail data from any double-buffered module pair and routes it to the DCS. Dynamic data steering is assumed for all multiplexers along with asynchronous control of buffer levels A and B, i.e., both half-duplex memory banks. Figure 6 also shows parity generation and detection blocks, which will be described later in this Section.

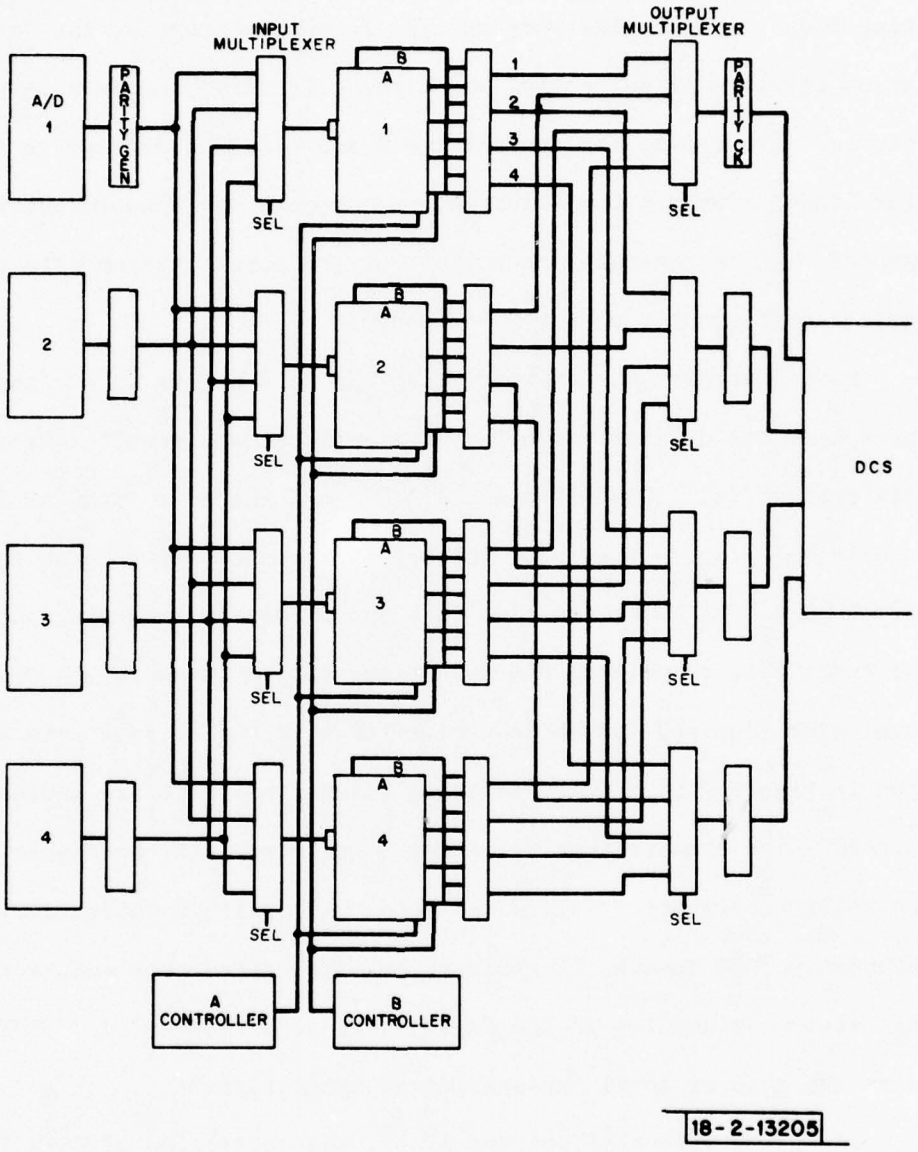


Fig. 6. Input buffer system 1: four channels, double-buffered.

Figures 7 and 8 show data streams and module activity for three typical operating modes, and provide some insight into the nature of the data flow, the allocation of time for collecting and dispensing data, the necessary data steering and duty cycles for controllers A and B, which each drive four modules. The time lines shown are for illustrative purposes, and should not be strictly interpreted. It is generally true that for the actual system data sets will not be supplied contiguously to the DCS by the IB.

Mode 1 permits the collection of 32K data points on two input channels, and the subsequent delivery of eight 8K transform data sets to the convolver as shown in Figure 7(a). Channel 3 and 4 A/D's are inactive, and the 8K data sets are interleaved (though they need not be) in terms of their input channel of origin. Figure 8(a) shows the time line approximately to scale, indicating that T_i, the total time to collect 64K points across two channels, is equal to T_o, the total time required for the convolver to absorb that same data when formatted as shown in Figure 7(a). The underlying assumptions here are dedicated A and B controllers which require that all A-level write activity be completed before the controller roles are reversed and level B is written while level A is read asynchronously. Of course, if one were willing to dedicate additional controllers, the data stored in modules 1A and 2A could be accessed earlier. System 2 later considers the case of total non-sharing of controllers.

In mode 2 (Figures 7(b) and 8(b)), the collection of data is much the same way as in mode 1, though each internal module configuration is changed from 2 to 1 for proper formatting upon output. Input and output time lines remain unchanged, while buffer module output rail switching via the output multiplexer is necessary.

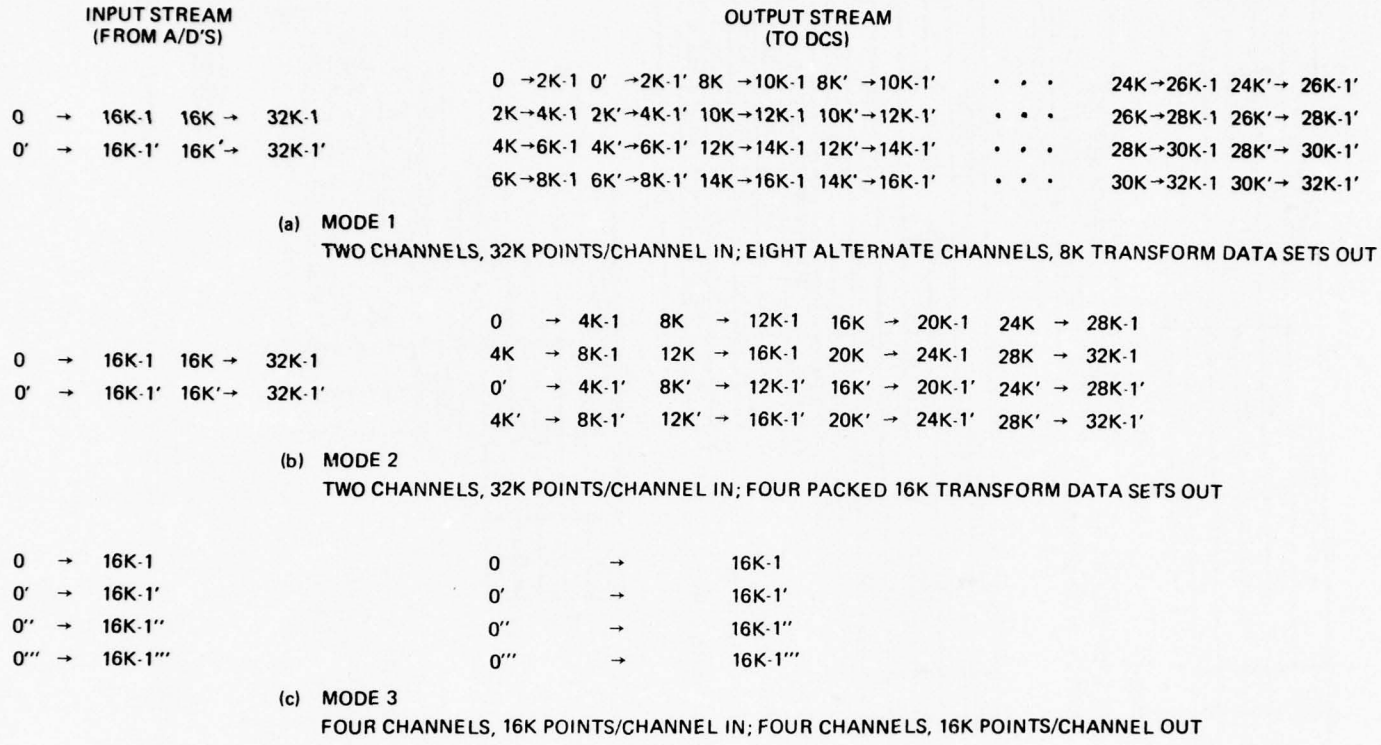


Fig. 7. System 1 data streams for 3 typical modes.

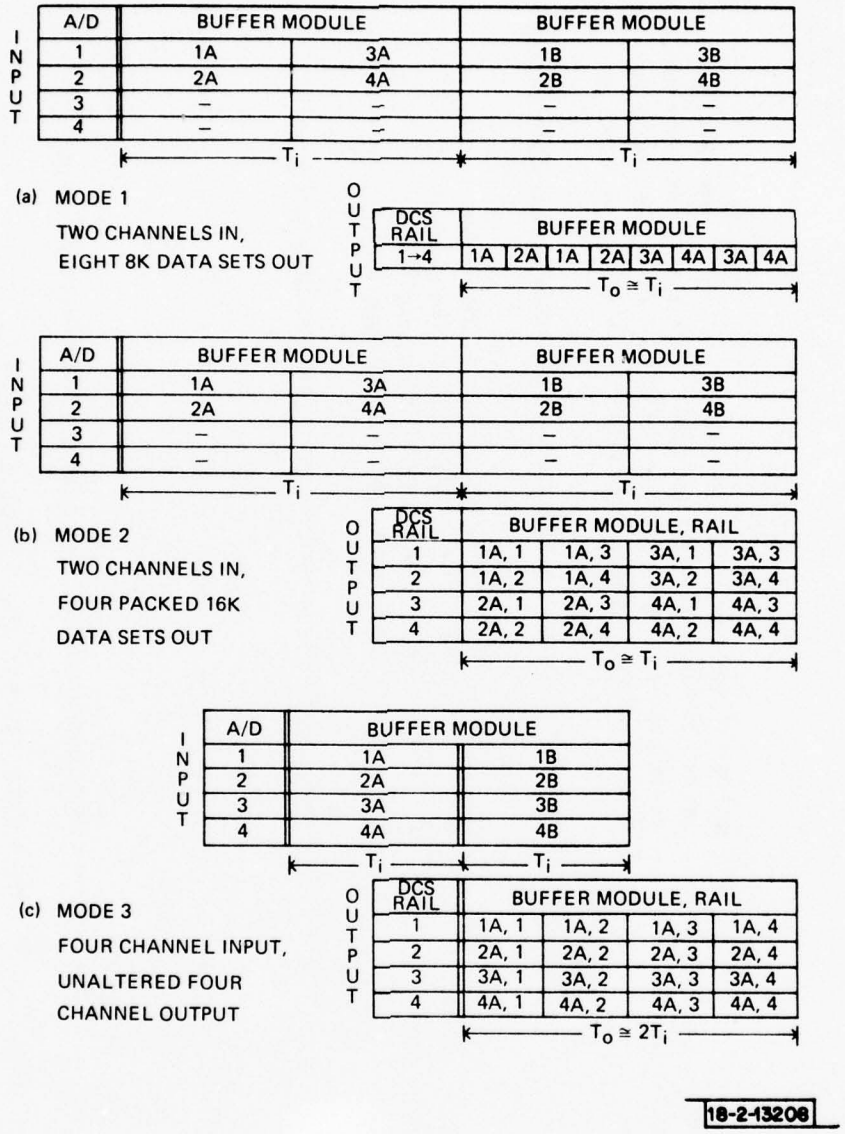


Fig. 8. System 1 memory module activity for modes 1, 2 and 3.

Mode 3 (Figures 7(c) and 8(c)) has the special property of delivering output data in exactly the same order in which it was stored. This feature is useful for playback or non-real time DCS processing.

B. Triple-Buffered System

Figure 9 shows a three-channeled input buffer design (System 2) which utilizes nine 16K x 24 buffer modules for triple-buffering. By the addition of individual module controllers and increased parallelism in both input and output data steering, System 2 not only provides triple-buffered, non-overlapped performance, but meets the requirements for overlapped data access, as we will see in the next section.

The same memory speed conventions apply to System 2 as for System 1. Referring to Figure 9, the general notion of employing input and output multiplexers still applies, although the design now includes an auxiliary channel. With appropriate control to avoid contention for modules, System 2 can support simultaneous traffic from two input or output devices. The multiplexers shown in Figure 9 should be considered "virtual multiplexers" in the sense that data buses could be used to minimize the hardware size of such a structure. With respect to control, individual controllers mean increased generality. For the non-overlapped, triple-buffered cases to be discussed here, however, it is useful to consider all controllers associated with levels A, B and C to be slaved together in groups of 3 modules.

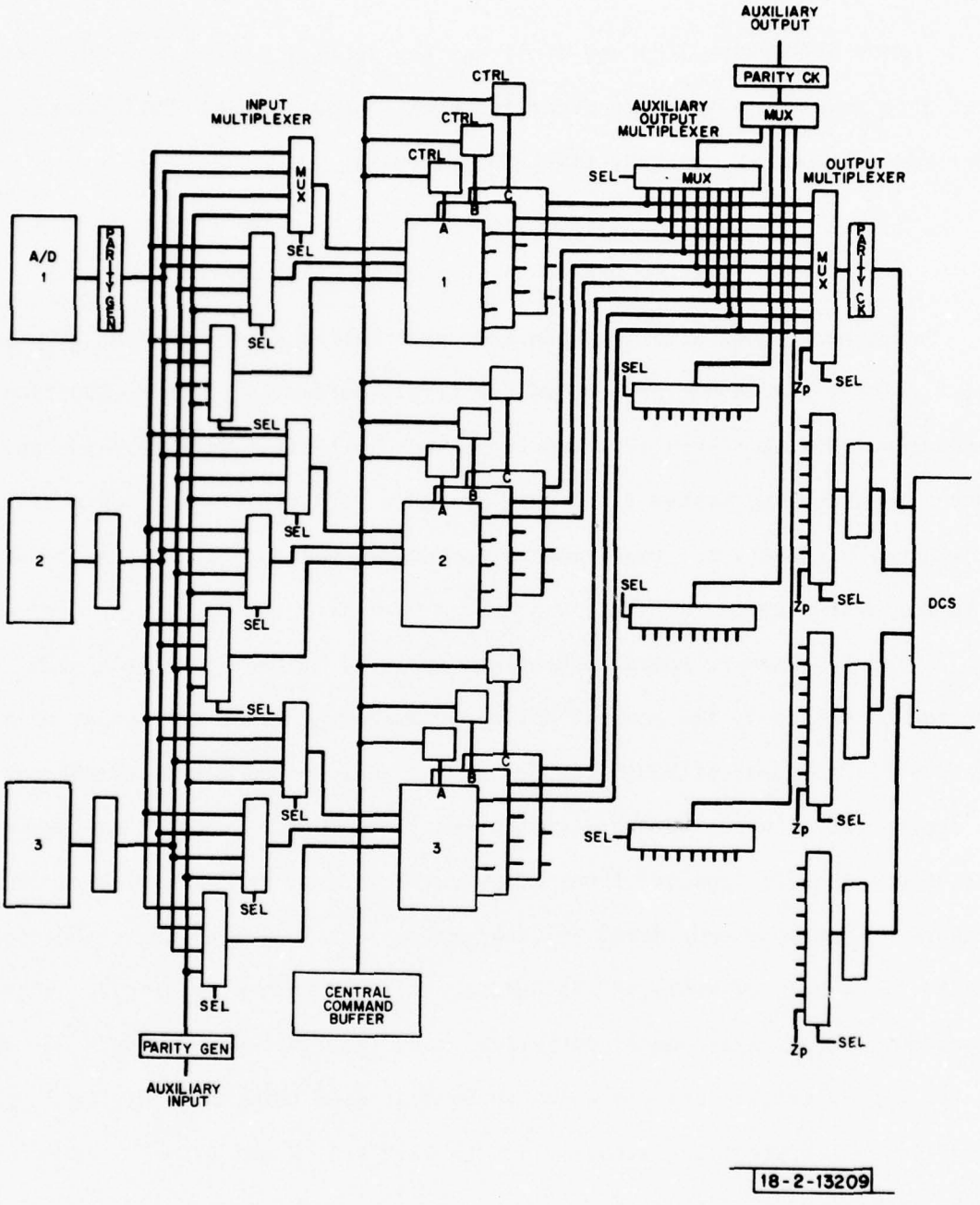


Fig. 9. Input buffer system 2: three channels, triple-buffered.

A three-channel IB system design which delivers data to a four-rail convolver imposes one additional requirement on the structure of Figure 9. When packing three-channel data, one DCS input rail must be inhibited or "zero padded". The "ZP" terminal on the output multiplexer of Figure 9 exists for this reason. Figures 10 and 11 show data streams and module activity for three typical operating modes, including a packed-data case.

Mode 1 is the collection of 48K data points on a single input channel, and the subsequent delivery of three 16K transform data sets to the convolver. Only the channel 1 A/D is active and, as shown in Figure 11, strict non-overlapping of data, starting with an empty buffer as shown, implies that no level A data may be offloaded until all level A writing has ceased. Of course, any stored data in levels B or C could be read concurrently with the filling of level A. Note that readout to the convolver requires half the data acquisition time.

Mode 2 is the first of two three-channel cases, and uses module configuration 2 to format six 8K transform data sets. Although the 8K sets are ordered sequentially according to input channel, consecutive 8K sets/channel are easily obtainable. No module rail switching is required.

Mode 3 is a three-channel, packed data mode which supplies four 16K transform data sets, and uses the zero pad feature discussed earlier.

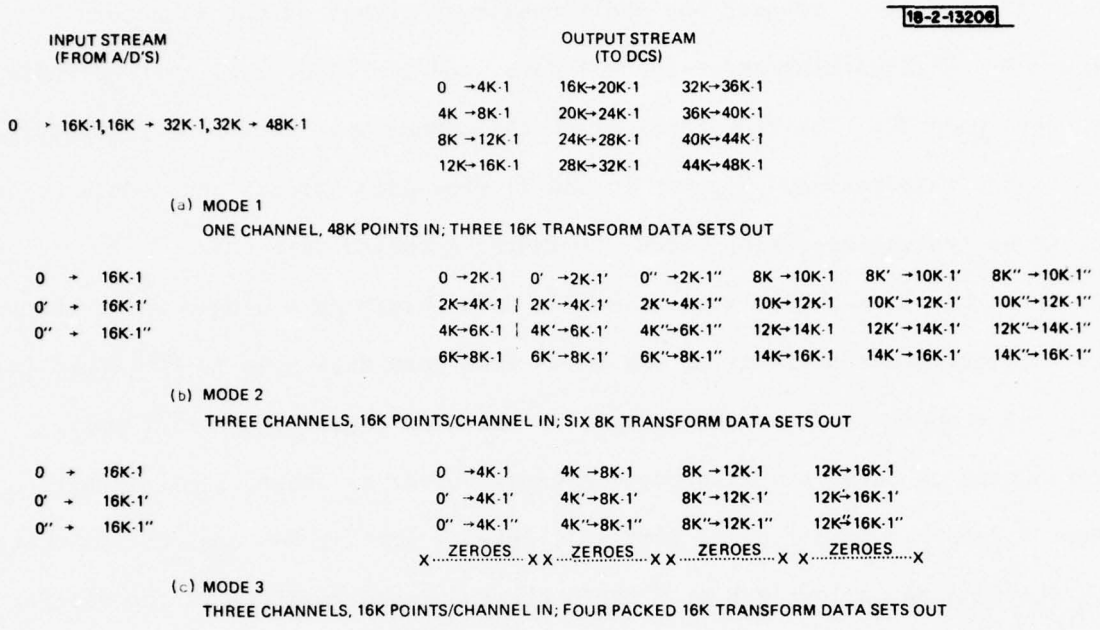


Fig. 10. System 2 data streams for 3 typical modes.

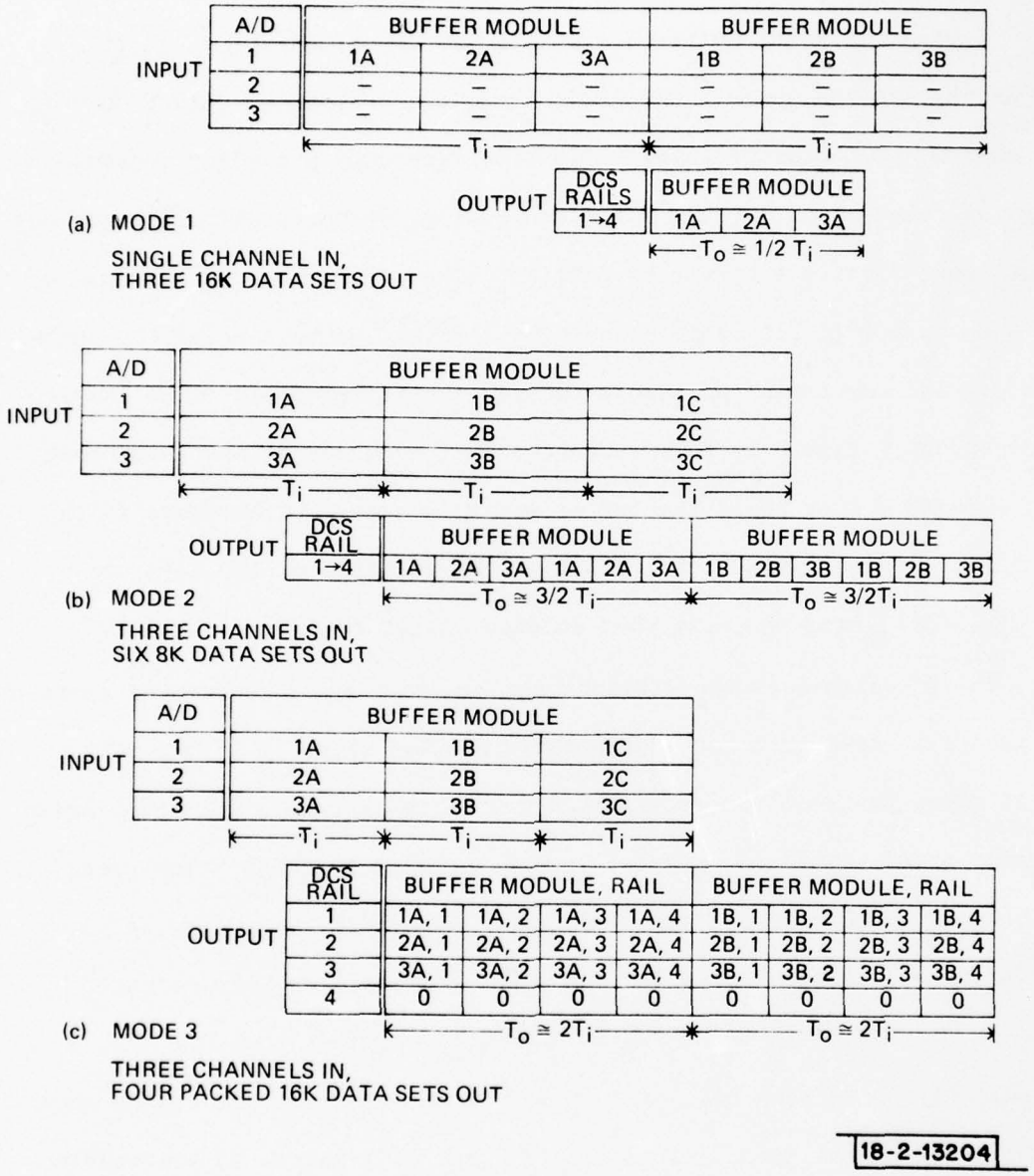


Fig. 11. System 2 memory module activity for modes 1, 2 and 3.

C. Multi-Overlapped System

Beyond the multi-buffered memory modes described above, there exists an application wherein the IB is used as a continuous data recorder at the system A/D converter rate, while simultaneously providing overlapped sets of the output data for all-range processing, in contrast to designs for non-overlapped data sets discussed earlier. Two alterations in System 2 of Figure 9 would permit all-range processing. First, we must relax the somewhat artificial constraint of slaving module controllers, and allow them to run independently (though in many instances they need not all be autonomous). Second, we must either build the buffer memory system with hardware which will support single-channel input rates in the vicinity of 120 Ms/s, or permit the DCS to stop and wait when IB data is not ready.

The selectable overlap feature may be realized by storage of certain data segments in more than one of the three modules shown in Figure 12. Figure 13 shows the required module read and write intervals for 25% overlap and 50% overlap cases. Some fundamental issues relating to equilibrium speeds for continuous throughput, amount of data overlap and multiplicity of memory modules require discussion. First, let us assume that the IB output data rate (i.e., the DCS input rate) will be a constant 120 Ms/s, which is equivalently 30 Ms/s on each of 4 rails. From Figure 13, it can be seen that some overlap sizes (>50%) lead to efficient utilization of all modules (all busy all the time), whereas overlaps of less than 50% have intervals during which no activity occurs. Such gaps must nonetheless be present to maintain appropriate redundant writing and continuous output streams.

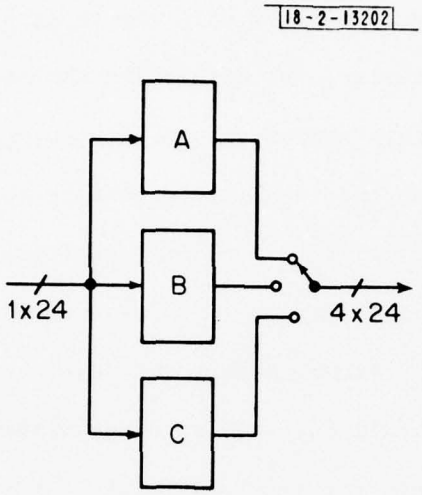


Fig. 12. Memory module configuration for continuous throughput.

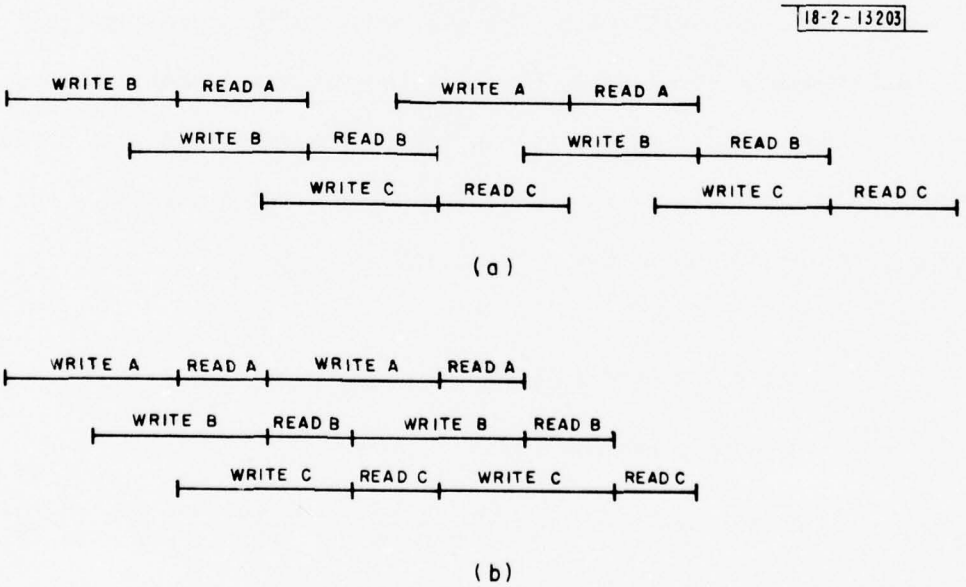


Fig. 13. Memory module read and write time intervals.

It should also be apparent that overlapping of output data while maintaining continuous throughput requires that the input be slowed by some factor related to the amount of overlap. We define the amount of overlap here to be the number of like data points stored redundantly in any two modules. As the amount of overlap increases, the input speed must be further slowed. Specific examples of data formatted for 4K, 8K or 16K transforms, and having various amounts of overlap, are shown in Figure 14. Data storage for these three transform sizes requires that individual memory modules be operated in configurations 1, 2 and 3, respectively. Full utilization of available memory storage capacity is required when storing data for 16K transforms, whereas 8K and 4K transforms require 50% and 25% capacity, respectively.

The maximum amount of overlap attainable is a direct function of the number of memory modules constituting the all range buffer configuration. While Figure 12 shows a module count of 3, which is the minimum number required, we could employ as many modules as are within the system, provided that performance would improve and that the necessary data and control paths existed. For any given module, let the following definitions apply:

T_W = write time per data set

T_R = read time per data set

L = normalized overlap = $\frac{\text{number of points common to two modules}}{\text{number of data points per data set}}$

N = number of buffer modules

18-2-13201

INPUT DATA PATTERN	OUTPUT DATA PATTERN
0→4K-1,4K→8K-1,8K→12K-1, ...	0 →1K-1,2K→3K-1,4K→5K-1,6K→7K-1, ...
	1K→2K-1,3K→4K-1,5K→6K-1,7K→8K-1, ...
	2K→3K-1,4K→5K-1,6K→7K-1,8K→9K-1, ...
	3K→4K-1,5K→6K-1,7K→8K-1,9K→10K-1, ...

(a) 4K TRANSFORMS, 50% (2K) OVERLAP

INPUT DATA PATTERN	OUTPUT DATA PATTERN
0→8K-1,8K→16K-1,16K→24K-1 ...	0 →2K-1,5K →7K-1, 10K→12K-1,15K→17K-1, ...
	2K→4K-1,7K →9K-1, 12K→14K-1,17K→19K-1, ...
	4K→6K-1,9K →11K-1,14K→16K-1,19K→21K-1, ...
	6K→8K-1,11K→13K-1,16K→18K-1,21K→23K-1, ...

(b) 8K TRANSFORMS, 37½% (3K) OVERLAP

INPUT DATA PATTERN	OUTPUT DATA PATTERN
0→16K-1,16K→32K-1,32K→48K-1, ...	0 →4K-1, 12K→16K-1,24K→28K-1,36K→40K-1, ...
	4K →8K-1, 16K→20K-1,28K→32K-1,40K→44K-1, ...
	8K →12K-1,20K→24K-1,32K→36K-1,44K→48K-1, ...
	12K→16K-1,24K→28K-1,36K→40K-1,48K→52K-1, ...

(c) 16K TRANSFORMS, 25% (4K) OVERLAP

Fig. 14. Representative data patterns for a 3-module, single-input, all range buffer configuration.

A simple examination of several timing cases such as those shown in Figure 13 indicates the following:

(1) $T_W = T_R / (1-L)$

(2) $T_R / T_W = 1 / (N-1)$ at maximum overlap

(3) $L_{max} = (N-2) / (N-1)$

Equation (1) indicates that module read and write time intervals are directly related to the amount of overlap, and that smaller overlaps permit memory write speeds to approach read speeds. A limit is set on writing speed under maximum overlap conditions as shown in Equation (2) based on the number of modules in use. Equation (3) stems from (1) and (2), and relates maximum overlap to module count. Figure 15 summarizes the above properties, and shows that extending the module count beyond 3 for any transform size extends the amount of achievable overlap, but with commensurate slowing of the input data rate.

From a radar standpoint, the transform size and time-bandwidth product determine the required amount of data overlap. Fortunately, from a hardware standpoint, one would ordinarily expect that all-range processing entails the collection of data sets having a size much larger than their associated TW product, meaning that the normalized overlap would be small. This fits nicely into the scenario of Figure 15, where small overlaps permit a small module count (usually 3), and the smallest input rate decrease.

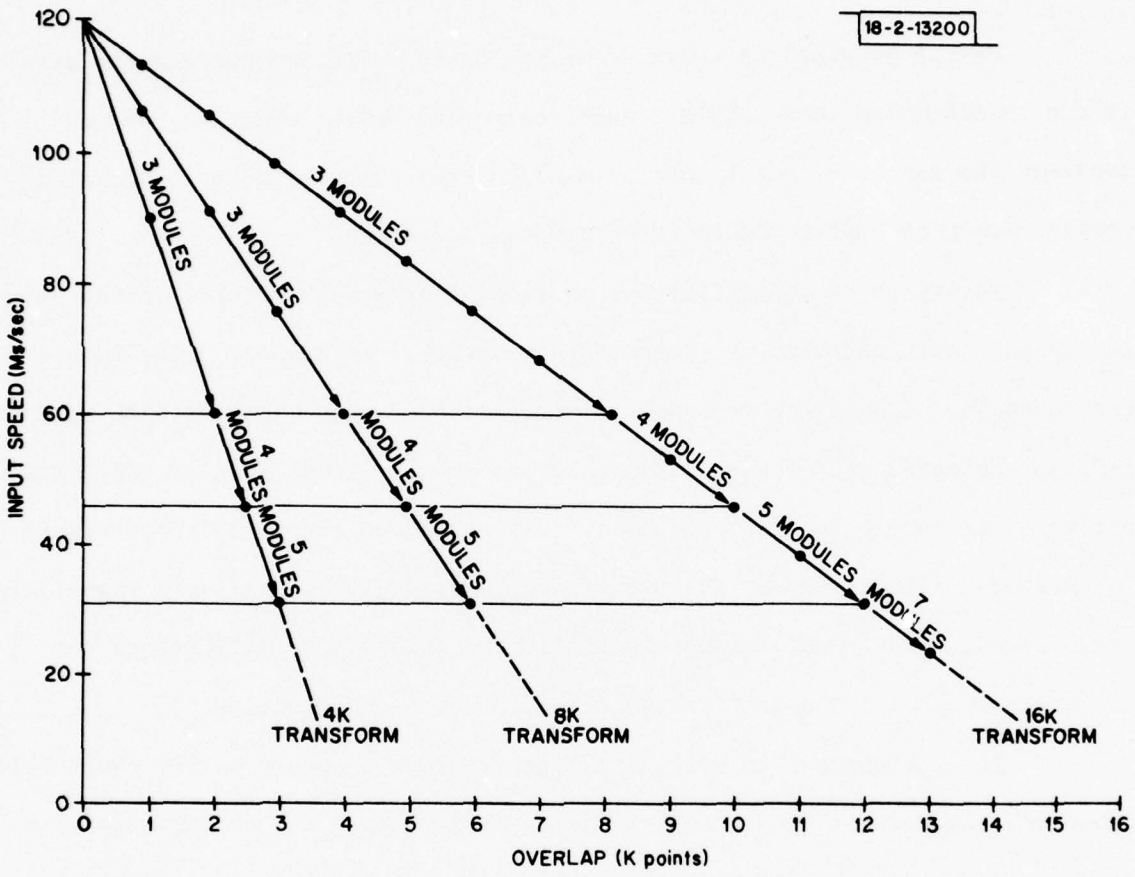


Fig. 15. Single-channel buffer input vs. amount of overlap for continuous throughput (all range) operation.

D. Parity

Having provided an overview of the basic buffer module and the ways it can be assembled into complete multi-channeled buffer memories, we now consider the issues of parity and command/control formats prior to detailed consideration of buffer module structure and operation.

Parity check capability for semiconductor memory modules of the size and speed considered here is recommended practice. In general, this involves the appending of an extra redundant binary bit onto each incoming data word to form an augmented word whose total number of 1's is always even or odd. Odd parity is preferred for two reasons. First, the zero bit usually represents the absence of a signal; hence, the use of odd parity ensures that some signal will be obtained, and second, for even word lengths, odd parity will detect the situation where all 1's or 0's have occurred due to malfunction.

It is important to note also that although a simple parity check will detect a single error in a group of digits of any size, the probability of a double error goes up as the length of the checked group increases. Thus, whether we check our module words en masse or test smaller groups of bits, some optimum span of bits must be chosen which is consistent with an allowable error limit.

More specifically, we have chosen to size the IB designs presented here so that 11-bit complex data (11 bits real, 11 bits imaginary) could be accommodated, with 1 parity bit to be allocated for each 11-bit byte.

Thus, with parity storage, the total buffer module word length is 24 bits. Both Figures 6 and 9 illustrate the positioning of parity generation and parity checking logic. Note that the parity logic spans not only the storage modules, but the input and output multiplexers as well.

E. Command Formats

The command/control issues for a buffer memory system, such as System 2 (a 3-channel, triple-buffered system), shown in Figure 9, are introduced here in a general way. Analysis in depth of the requisite IB control states is an appropriate part of a detailed design, but beyond the scope of this report.

Control of the complete System 2 can be described in terms of a central command buffer which passes arguments to nine individual module controllers, each capable of asynchronous read or write operations with respect to other modules, and which retains other parameters associated with logic mutually shared by all modules (e.g., the steering of input and output devices).

Three kinds of command/control signals are envisioned here, and are termed 1) initialization levels, 2) flags and 3) triggers. Initialization levels are supplied to the IB prior to the beginning of memory data transfer. Triggers and flags, on the other hand, are initiators and indicators of events occurring during IB operation. A list of possible command/control signals is given below in Table 1, together with brief references to their function.

Figure 9 shows conceptually the interconnection paths among the A/Ds, IB, DCS and auxiliary devices. This arrangement permits easy module reassign-

TABLE 1

COMMAND/CONTROL SIGNALS FOR A
3-CHANNEL TRIPLE-BUFFERED MEMORY SYSTEM

INITIALIZATION LEVELS	# BITS	DESCRIPTION
Buffer Channel Select	2	Selects active IB module(s) across all channels
Buffer Level Select	2	Selects active IB module(s) by channel
Input Device Select	2	Selects A/D or Auxiliary inputs
Transform Size	2	4K, 8K, 16K information used chiefly for A/D and DCS transfers
Output Device Select	2	Selects DCS or Auxiliary outputs
Pack Select	1	Refers to IB outputs only, alters module output steering; doesn't change output device
Data Suppress, Start Pt, End Pt	(1,4,4)	Used to blank fourth DCS input channel when packing, address to nearest 1K, for IB reads only
Read/Write	1	Programs memory module(s) to read or write when triggered
Start Pt, End Pt, # Data Sets	(4,4,2)	Specifies start and end addresses to within nearest 1Kth interval of total module address space
FLAGS	**# BITS**	**DESCRIPTION**
IB Output Ready	1	Indicates to output device that IB is ready to deliver data, but module isn't necessarily full
IB Input Ready	1	Indicates to input device that IB is ready to accept data, but module isn't necessarily empty
TRIGGERS	**# BITS**	**DESCRIPTION**
IB Output Request	1	
(a) DCS Input Request		Requests IB→DCS data transfer
(b) Aux Input Request		Requests IB→Aux data transfer (Data reformat required)
IB Input Request	1	
(a) A/D Start		A/D→IB transfers if IB ready; else command is aborted
(b) Aux Output Request		When IB is ready, Aux Output Req begins Aux device→IB transfer (data reformat required)

ment, and supports up to two simultaneous asynchronous reads or writes per channel. Recall that the 31 bits of initialization previously listed are to be relayed to all 9 modules in turn. Thus, for concatenating operations on successive data sets, additional initialization data must first be routed through the central command buffer and then be queued in fast stacks within the appropriate module controllers. This permits real-time reconfiguration at the module level.

For most of the control states outlined here, the Signal Processor Control (see Figure 2), which is well up in the control hierarchy, mediates any contention among devices vying for IB access. If we say that the A/D to IB to DCS path constitutes the primary data channel, then the Signal Processor Control will decide when auxiliary devices may have access to the IB.

IV. DETAILED MEMORY MODULE DESCRIPTION

A 1-bit slice of the basic buffer module, described in Section II, is shown in Figure 16 below. Recalling that either one, two or four input channels may be active, the front end functions as a data rate reducer prior to depositing data within the memory array via the input steering section. Alterable input and output data paths permit stored data to be retrieved from the memory array in several different ways. Independent addressing and control states are required for modules being written while others are being read; i.e., independent controllers are required for asynchronous operation. For the large multi-buffered structures described earlier, address/control logic can be shared among those modules which are either read or written at the same time.

Some conventions are necessary with respect to input data rates as a function of the number of active input channels. Let us define f_1, f_2 and f_4 as the input sample (word) rates for one, two and four channel operation, respectively. Based upon hardware experimentation to be discussed later, the expected values of f_1, f_2 and f_4 are 190 Ms/s, 95 Ms/s and 47 megasamples/sec.

One and two channel rates in excess of 60 Ms/s permit the buffer system to keep pace with a planned 60 Ms/s A/D converter. The four-channel configuration of the basic module would constrain the input rate to 47 Ms/s per channel. However, any realistic system would use a combination of four to eight modules, each with only one (or at most two) active inputs, thus accommodating input rates of at least 95 Ms/s. The experimentally anticipated rate of f_1 = 190 Ms/s is clearly the most stressing, and, to the extent that the proposed buffer hardware can support it, future A/D converter rates up to 190 Ms/s could be accommodated. The confidence level in values for f_{1max} will be discussed further in connection with experimental hardware results.

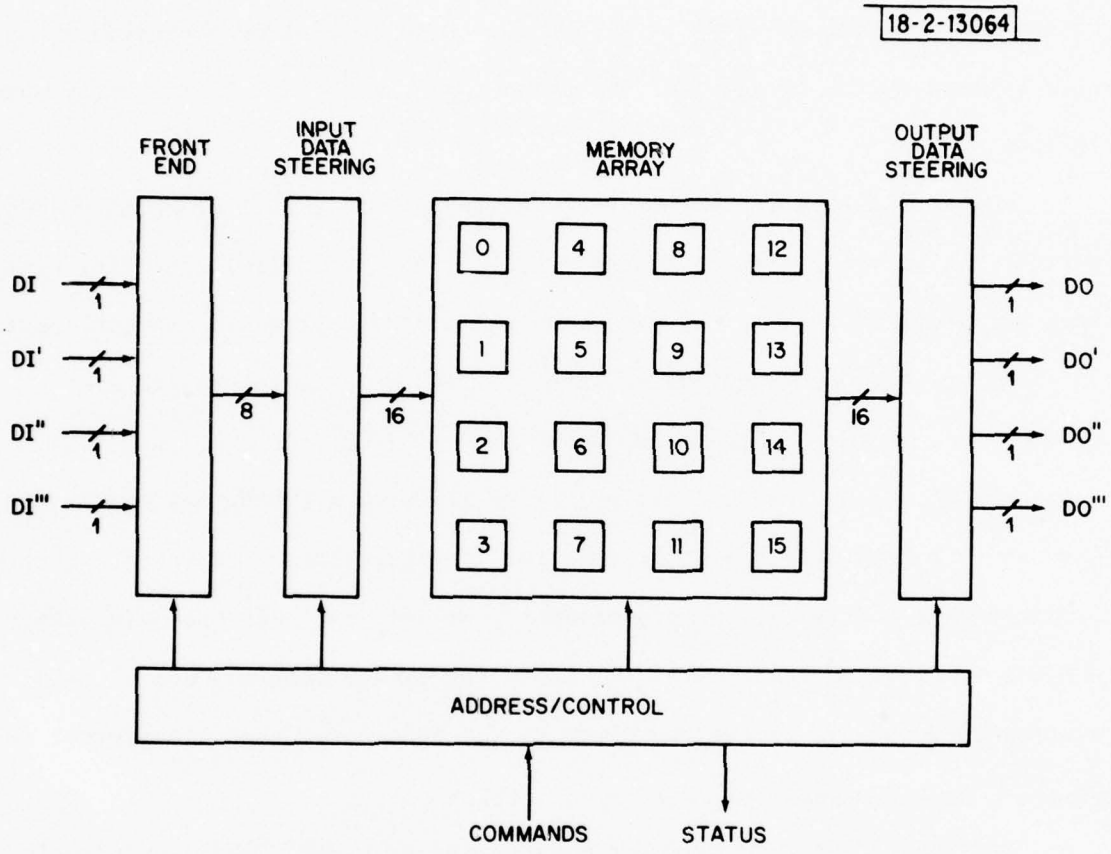


Fig. 16. Block diagram of basic buffer memory module (1-bit slice).

A. Front End

The IB front end is simply a serial-to-parallel converter followed by a holding register as shown in Figure 17. Data multiplexers provide a choice between serial or parallel operation, along with serial data entry point selection.

For single-rail inputs (module configurations 1, 2, 3 of Section II), the front end is configured as a single 8:1 serial-to-parallel converter with output register, and delivers 8-bit words at $f_1/8$ megawords/sec. to the input data steering section. Input data rates of f_1>60 Ms/s represent the most stressful case in terms of hardware speed, and will be discussed in detail. For the moment, then, let us consider Figure 17 to be a conceptual model only. Figure 18 is a front end timing diagram for configurations 1, 2 and 3, wherein the converter clock (SCLK) has a periodicity of $1/f_1$ ns. All data are numbered according to order of occurrence. As indicated by the diagram, serial data are transferred upward in the converter at the SCLK rate; then transferred to the output register under parallel clock (PCLK) control.

Two input rails are used for configurations 4 and 5, and accordingly the front end is arranged as a pair of 4:1 serial-to-parallel converters with parallel output registers. On a per-rail basis, the input data rate is now reduced to f_2 Mwords/sec.; i.e., SCLK has a periodicity of $1/f_2$ ns. However, the net front end data throughput rate becomes (2 channels) x (f_2 Mwords/sec/channel) = $2f_2$ Mwords/sec. Figure 19 shows the timing relationship for configurations 4, 5 and 6. Note that all data are labelled to denote their sequential nature as well as input channel rail of origin.

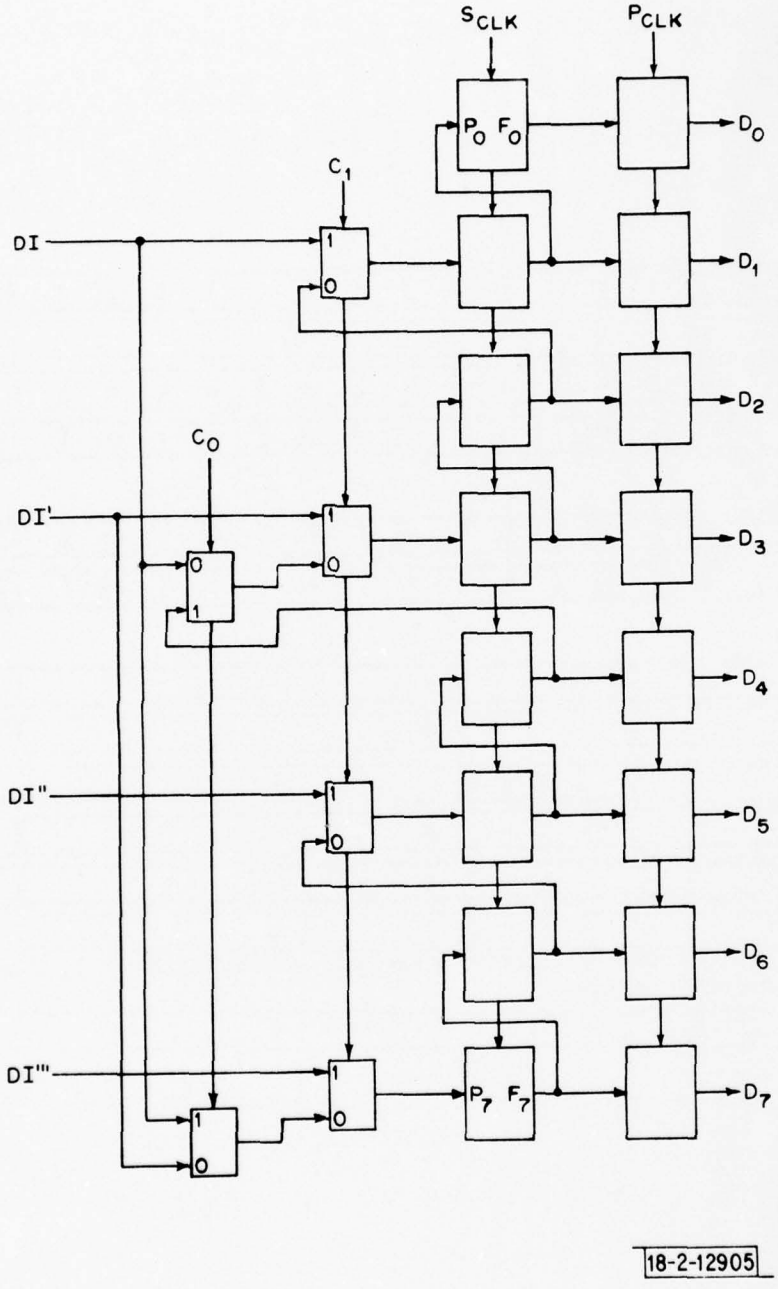


Fig. 17. Buffer memory front end.

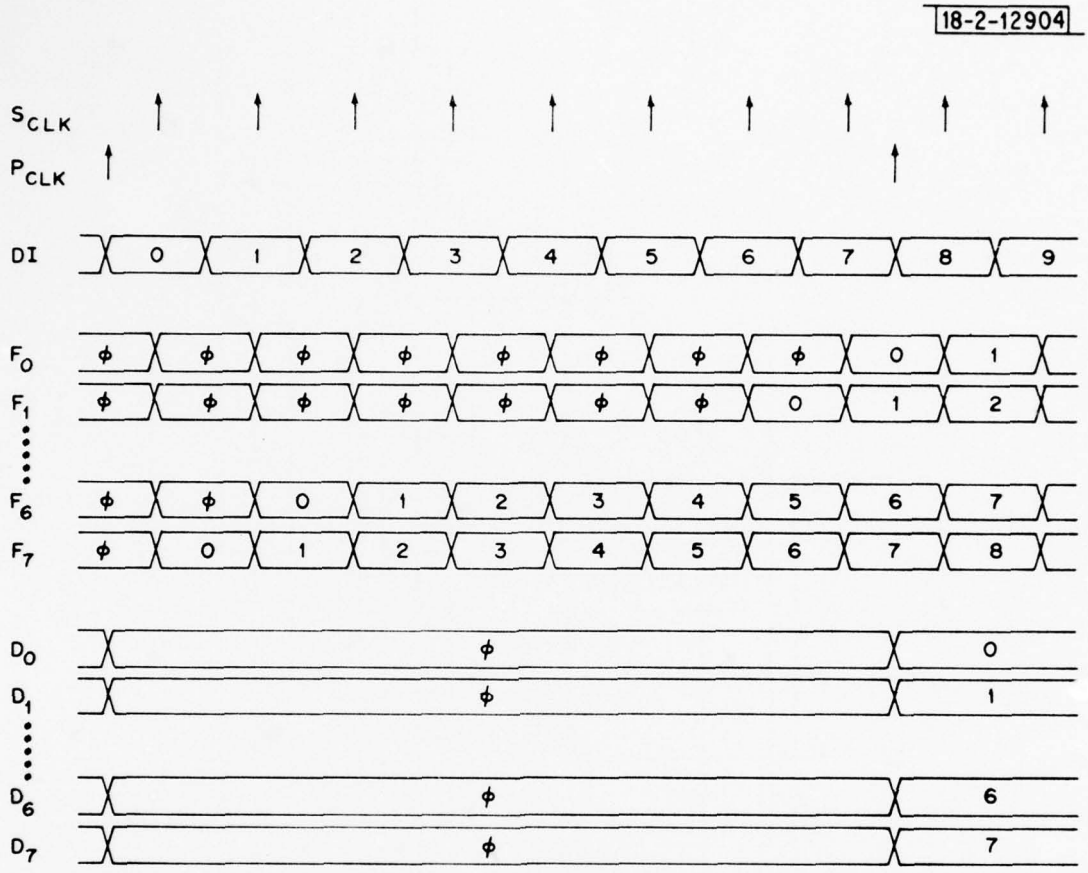


Fig. 18. Front end timing diagram, configurations 1, 2 and 3.*
* ϕ = don't care.

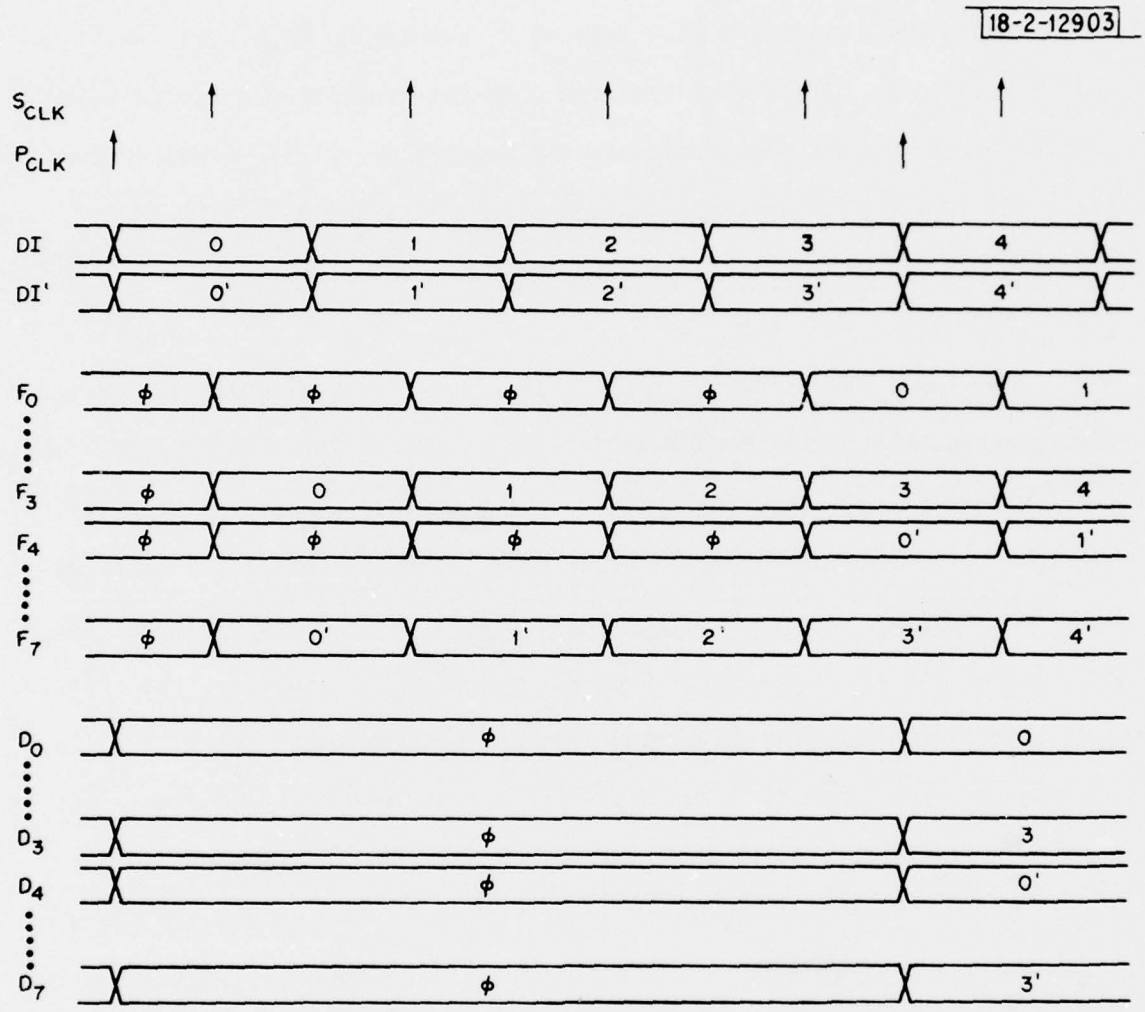


Fig. 19. Front end timing diagram, configurations 4, 5 and 6.

For configurations 7 through 9, the front end must accept data on 4 input rails, each operating at a rate of f_4 words/sec. Four 2:1 serial-to-parallel converters perform the required data rate reduction prior to being strobed into the output register, which is updated every $2/f_4$ nsec. Figure 20 depicts the front end timing for configurations 7, 8 and 9.

B. Input Data Steering

The input data steering section is simply a set of multiplexers which are interposed between the front end data register and the memory array. While several mode-dependent steering options exist, all are consistent with left bank/right bank, partial left bank/right bank, or upper bank/lower bank data routing strategies. Figure 21 denotes the available data paths between the front end and the storage array. Control word C7...C2 determines the steering format. As shown in Figure 22(a) and (b), buffer configurations 1, 2, 3, 5 and 6 require dynamic control for alternate column interchanging within either the left or right bank of the memory array. Configurations 8 and 9 do not require the transposing of upper and lower bank rows, but do require the dynamic end-for-end swapping of data within each row of the memory array as shown in Figures 22(d) and (e). Configuration 4 utilizes static data steering (Figure 22(c)), using what might be termed an interleaved left bank/right bank format. It should be noted that bank selection per se is a function of memory chip select logic. For example, while RAM 0 and RAM 8 may simultaneously receive data word D_o from the front end (Figure 22(a)), only one of the two memory elements will be written at any given time.

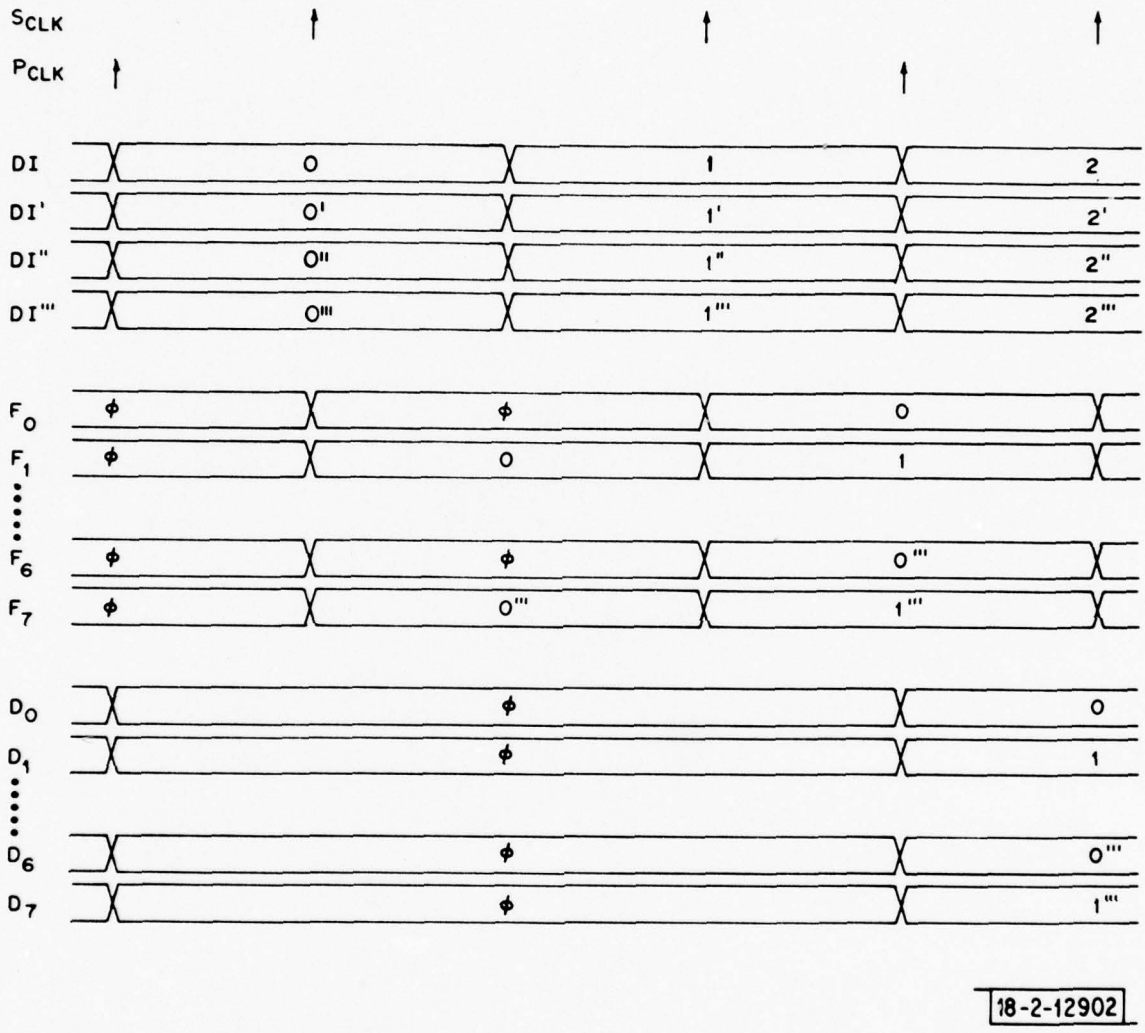


Fig. 20. Front end timing diagram, configurations 7, 8 and 9.

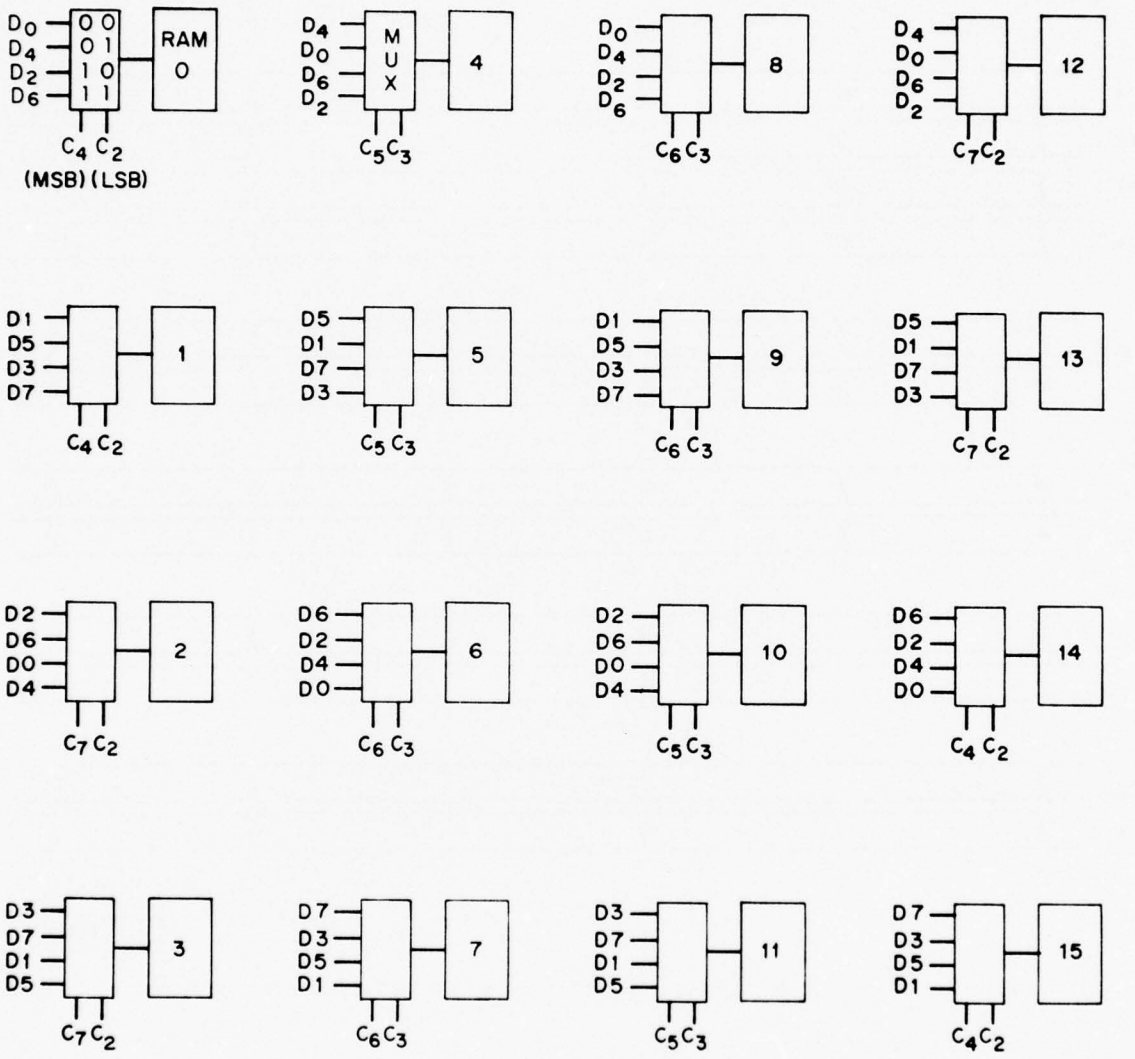


Fig. 21. Input data steering section.

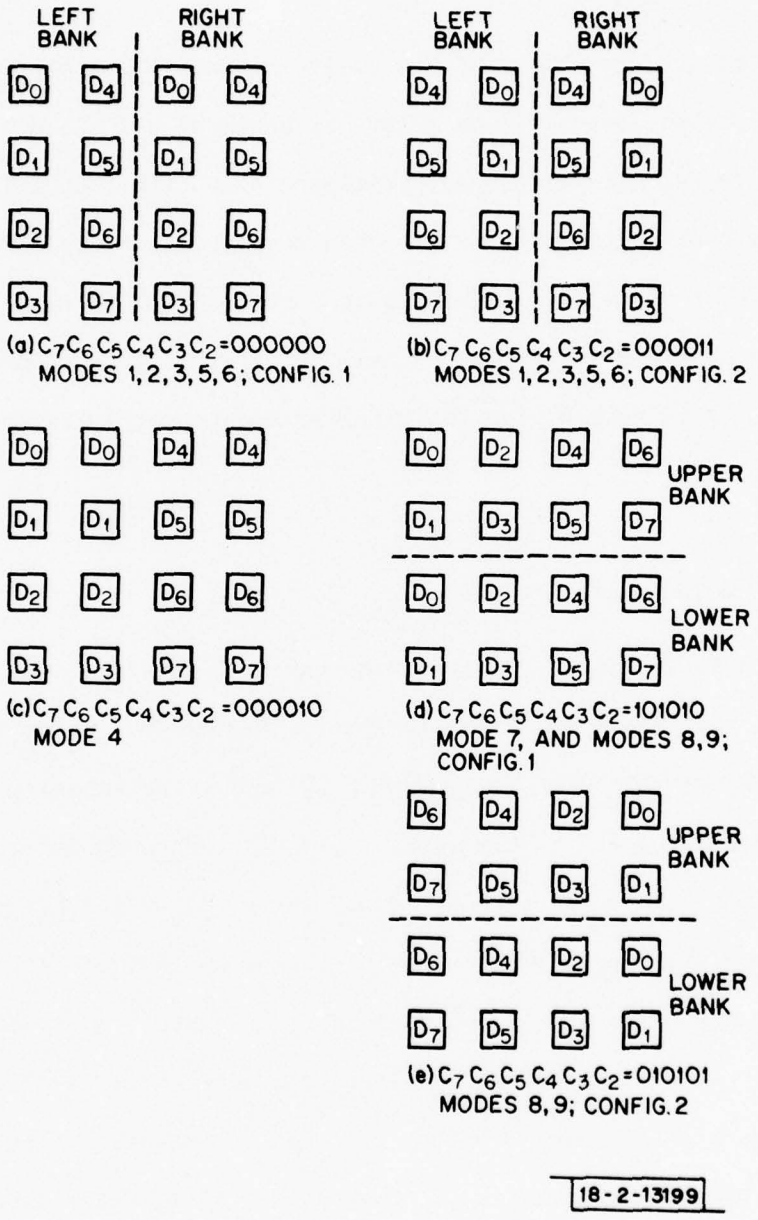


Fig. 22. Storage array input data connections.

C. Memory Array

The memory array is simply a collection of 1K x 1 bipolar random-access integrated circuits, organized on a per-bit basis as a 4 x 4 device array. As shown in Figure 22, data access for writing can be either on a left bank/right bank or upper bank/lower bank basis. When reading the array, access is made to groupings of four devices as indicated in Figure 23. A thorough examination of the necessary memory array control states (e.g., chip select, page select, page address, write enable) for the 9 buffer module configurations will be made in a later section.

D. Output Data Steering

The output data steering section has been designed with particular emphasis upon hardware simplicity and speed. Figure 23 denotes the grouping of semiconductor memory devices such that all necessary intergroup connections may be realized using 4:1 multiplexers. For all configurations, the positioning of data within the basic 16-element memory array places the burden of hardware complexity upon the input section, the payoff being that pipelined addressing may be used across each of the 4-RAM groups of Figure 23 to enhance buffer memory output speed. As will be later described, the wire-ORing of memory devices simplifies the explicit data steering problem, but requires that individual RAM contributions to group buses (N_0 through N_3) be mutually exclusive. Pipelined addressing in itself implies the existence of an address delay register stack, together with non-overlapping chip selects.

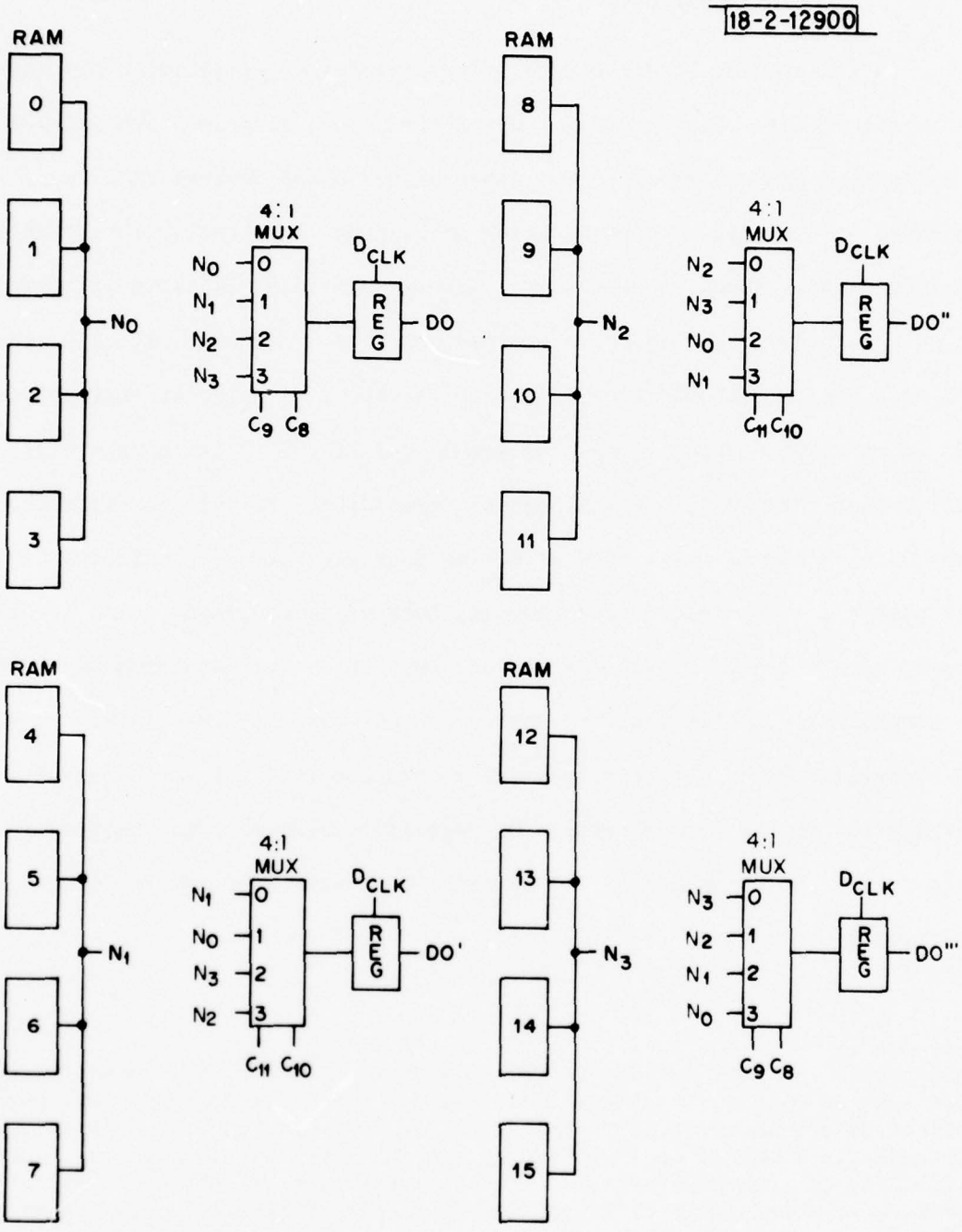


Fig. 23. Output data steering section.

E. Address/Control

A synchronous binary counter forms the basis for deriving all address and control states. Figure 24 is a simplified block diagram which introduces the write/read command (W/R), and a 4-bit binary coded decimal (BCD) configuration select word, M. For writing or reading, some configurations require different counter speeds than others. Memory addressing is shown in terms of chip select, page select and page address sections. For writing, chip select performs a bank select function, and enables the write pulse at appropriate RAMS. When reading, non-overlapping groups (4) of chip selects make fast memory output rates possible via address pipelining. For all configurations except 3, each RAM is envisioned as having four pages (mode 3 utilizes eight "half pages"). The strategy for choosing both write and read paging algorithms is based upon the need *ultimately to read out contiguous data from adjacent* RAMS sequentially, while avoiding memory contention. As shown in Figure 24, read addresses are offset in time using a register stack (i.e., pipelined) before being applied to the memory array. On-page addressing is binary sequential. Tables 2 and 3 below summarize the specific counter states for each configuration.*

---

* This report does not purport to describe every nuance of the control problem. Issues such as optimal phasing of data arrival for reliable input registering, or offset delays to correlate output steering control with data delayed due to RAM access times should be addressed by the final hardware designers. The availability of counter states and static mode-select states should make the generation of such status flags as "busy", "full", "1/2 full", etc. very straightforward.

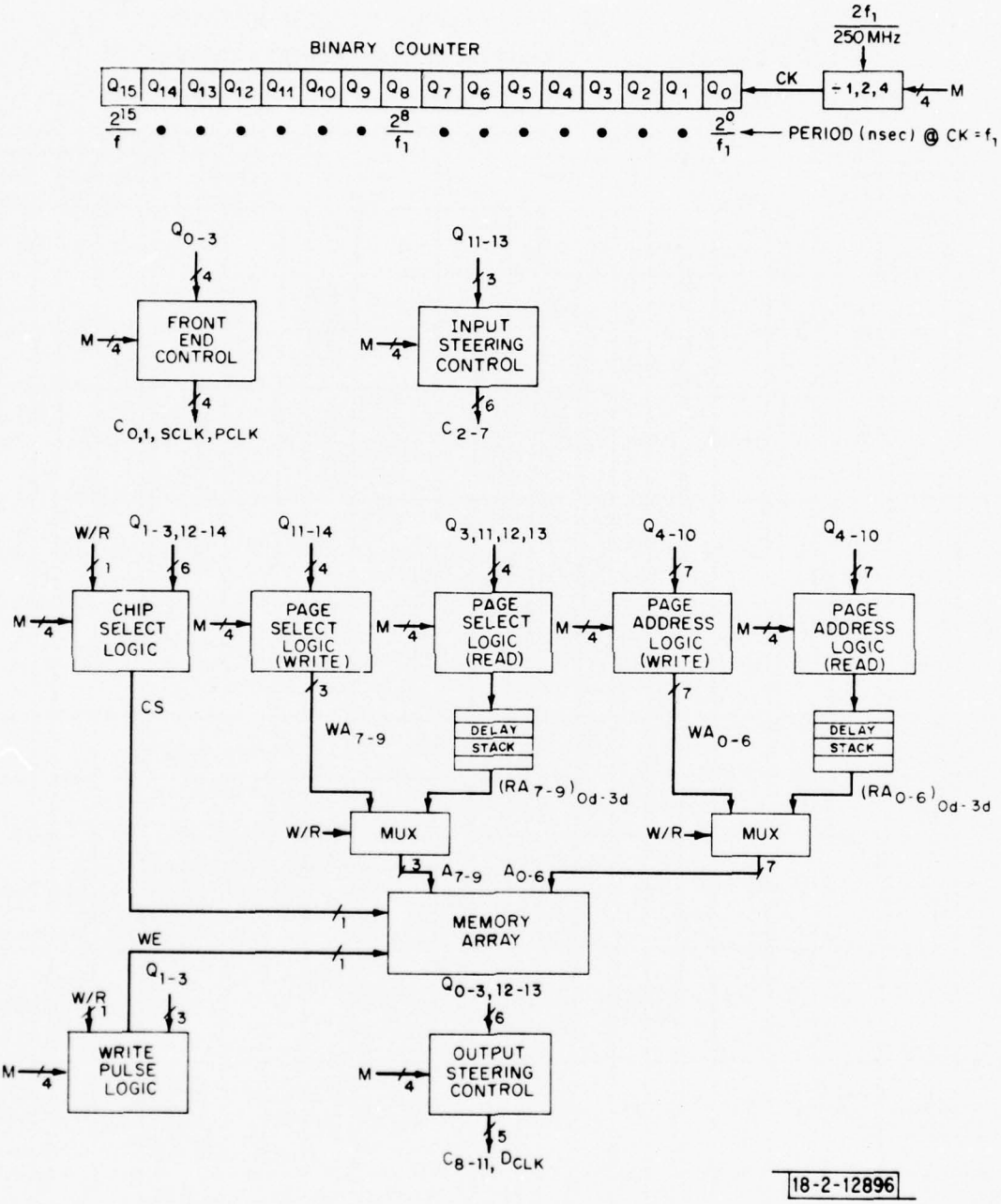


Fig. 24. Address and control section block diagram.

TABLE 2

BUFFER MODULE CONTROL STATES FOR WRITING

CONFIG	NUMBER OF INPUT CHANNELS	COUNTER SPEED SCALE FACTOR	FRONT END				INPUT STEERING						CHIP SELECT		PAGE SELECT			PAGE ADDRESS		WRITE PULSE
			PCLK	SCLK	C1	C0	C7	C6	C5	C4	C3	C2	RAMS	CS	RAMS	A9	A8	RAMS	A7-A0	WI
1	1	÷1	$\overline{Q3}$	Q0	0	1	0	0	0	0	Q13	Q13	0-7	$\overline{Q14}$	ALL	Q13	Q12	ALL	Q11-Q4	Q1⊙Q2
													8-15	Q14						
2	1	÷1	$\overline{Q3}$	Q0	0	1	0	0	0	0	Q12	Q12	0-7	$\overline{Q13}$	ALL	Q14 ⊕ Q13	Q12	ALL	Q11-Q4	Q1⊙Q2
													8-15	Q13						
3	1	÷1	$\overline{Q3}$	Q0	0	1	0	0	0	0	Q11	Q11	0-7	$\overline{Q12}$	ALL	Q14 ⊕ Q12	Q13	ALL	Q11-Q4	Q1⊙Q2
													8-15	Q12						
4	2	÷1	$\overline{Q3}$	Q1	0	0	0	0	0	0	1	0	0-3, 8-11	$\overline{Q14}$	ALL	Q14	Q13	ALL	Q11-Q4	Q1⊙Q2
													4-7, 12-15	Q14						
5	2	1	$\overline{Q3}$	Q1	0	0	0	0	0	0	Q13	Q13	0-7	$\overline{Q14}$	0-3,8-11	Q13	Q12	ALL	Q11-Q4	Q1⊙Q2
													8-15	Q14	4-7,12-15	$\overline{Q13}$	Q12			
6	2	÷1	$\overline{Q3}$	Q1	0	0	0	0	0	0	Q12	Q12	0-7	$\overline{Q13}$	0-3,8-11	Q12	Q14	ALL	Q11-Q4	Q1⊙Q2
													8-15	Q13	4-7,12-15	$\overline{Q12}$	Q14			
7	4	÷2	$\overline{Q2}$	Q1	1	ϕ	1	0	1	0	1	0	0,1,4,5, 8,9,12,13	$\overline{Q3}$	ALL	Q13	Q12	ALL	Q11-Q4	Q1⊙Q1
													2,3,6,7, 10,11,14,15	Q3						
8	4	÷2	$\overline{Q2}$	Q1	1	ϕ	$\overline{Q12}$	Q12	$\overline{Q12}$	Q12	$\overline{Q13}$	Q13	0,1,4,5, 8,9,12,13	$\overline{Q3}$	0-3	Q13	Q12	ALL	Q11-Q4	Q1⊙Q1
															4-7	Q13	$\overline{Q12}$			
													2,3,6,7, 10,11,14,15	Q3	8-11	$\overline{Q13}$	Q12			
															12-15	$\overline{Q13}$	$\overline{Q12}$			
9	4	÷2	$\overline{Q2}$	Q1	1	ϕ	$\overline{Q12}$	Q12	$\overline{Q12}$	Q12	$\overline{Q13}$	Q13	0,1,4,5, 8,9,12,13	$\overline{Q3}$	0-3	Q13	Q12	ALL	Q11-Q4	Q1⊙Q1
															4-7	Q13	$\overline{Q12}$			
													2,3,6,7, 10,11,14,15	Q3	8-11	$\overline{Q13}$	Q12			
															12-15	$\overline{Q13}$	$\overline{Q12}$			

TABLE 3

BUFFER MODULE CONTROL STATES FOR READING

* ADDRESS STATES PRIOR TO STACK DELAYS
** CONFIGURATION 3 HAS UNIQUE 8-PAGE ORGANIZATION

CONFIG	COUNTER SPEED SCALE FACTOR	CHIP SELECT		PAGE SELECT*				PAGE ADDRESS		OUTPUT STEERING				
		RAMS	CS	RAMS	A9	A8	A7	RAMS	A7-A0	C11	C10	C9	C8	PCLK
1	÷4	0,4,8,12 1,5,9,13 2,6,10,14 3,7,11,15	$\overline{Q1}\cdot Q2$ $Q1\cdot Q2$ $\overline{Q1}\cdot\overline{Q2}$ $Q1\cdot\overline{Q2}$	0→3,8→11 4→7,12→15	Q3 Q3	Q12 Q12		ALL	Q11→Q4	0	$\overline{Q2}\oplus Q3$	0	$\overline{Q2}\oplus Q3$	Q0
2	÷4	SAME	SAME	0→3 4→7 8→11 12→15	Q12 Q12 Q12 Q12	$\overline{Q3}$ $\overline{Q3}$ Q3 $\overline{Q3}$		ALL	Q11→Q4	0	$\overline{Q2}\oplus Q3$	0	$\overline{Q2}\oplus Q3$	Q0
3	÷4	SAME	SAME	0→3 4→7 8→11 12→15	Q12 Q12 $\overline{Q12}$ $\overline{Q12}$	Q11 Q11 Q11 Q11	$\overline{Q3}$ $\overline{Q3}$ Q3 $\overline{Q3}$	ALL	* *	0	$\overline{Q2}\oplus Q3$	0	$\overline{Q2}\oplus Q3$	Q0
4	÷2	0,4,8,12 10,5,9,13 2,6,10,14 3,7,11,15	$\overline{Q2}+\overline{Q3}$ $\overline{Q2}+\overline{Q3}$ $Q2+Q3$ $\overline{Q2}+Q3$	ALL	Q13	Q12		ALL	Q11→Q4	0	0	0	0	Q1
5	÷2	SAME	SAME	ALL	Q13	Q12		ALL	Q11→Q4	0	Q13	0	Q13	Q1
6	÷2	SAME	SAME	ALL	Q12	Q13		ALL	Q11→Q4	0	Q12	0	Q12	Q1
7	÷2	SAME	SAME	ALL	Q13	Q12		ALL	Q11→Q4	0	0	0	0	Q1
8	÷2	SAME	SAME	0→3,8→11 4→7,12→15	Q13 Q13	Q12 $\overline{Q12}$		ALL	Q11→Q4	$\overline{Q13}$	$\overline{Q12}$	Q13	Q12	Q1
9	÷2	SAME	SAME	ALL	Q13	Q12		ALL	Q11→Q4	Q13	Q12	Q13	Q12	Q1

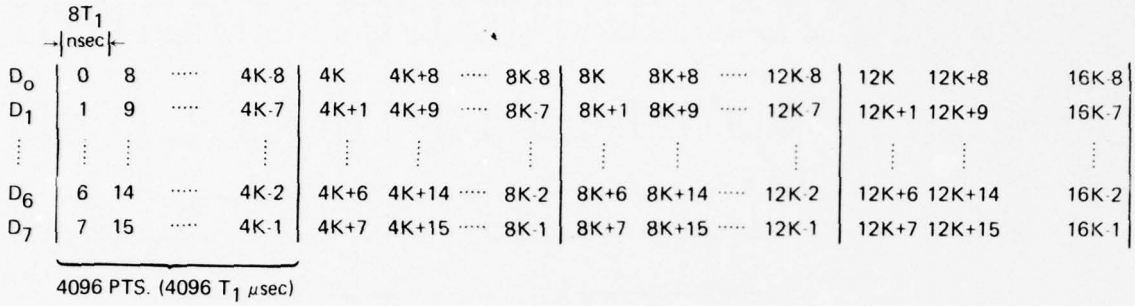
F. Buffer Module Configurations

This section completes the discussion of the basic buffer module structure by describing the nature of the data flow for all 9 configurations with the aid of memory load maps as summarized in Section II. Configuration 1 is described in extra detail since it is representative of the configurations used within the systems described in Section III. This discussion includes data stream and timing diagrams which are representative of the control and data activity found in other configurations.

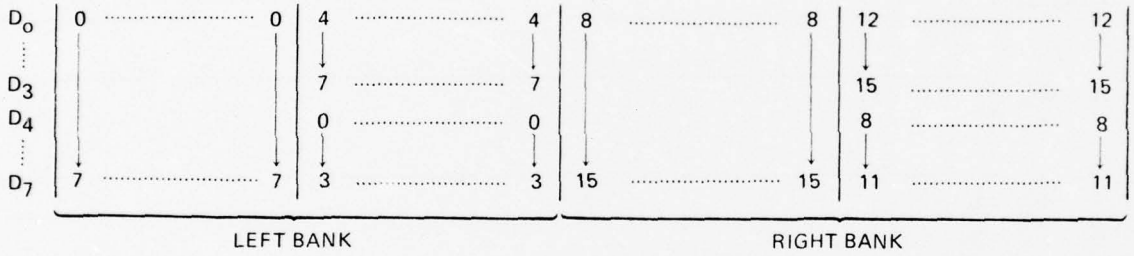
Configuration 1

Configuration 1 requires that the buffer memory front end operate at the single-rail input data rate ($f_1 = 1/T_1$), and supply an 8-bit datum to the input steering section every $8T_1$ nanoseconds. Figure 25(a) describes the front end data flow. The entire left bank (RAMS 0 through 7) is loaded before filling the right bank. As shown in Figure 25(b), an input steering change is required every 4096 T_1 ns. Address space for the 16 RAMS which comprise each 1-bit memory slice is divided into four pages. Figure 25(c) relates page selection during writing to input data activity. Within a given page, addressing is always binary sequential. Basic timing relationships for memory write activity are shown in Figure 26. The time interval shown is that for the first two write operations (i.e., the loading of the first 16 serial data points) within the left bank, and presupposes that the first front end S/P conversion has preceded the first write pulse.

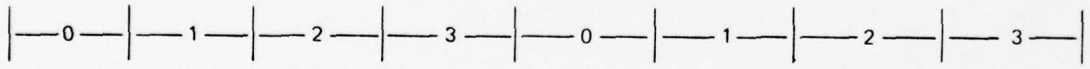
Figure 27 is the memory load map for configuration 1 and describes the way in which 16,384 sequential data points (words) reside in memory. The memory map notion becomes especially useful when examining the addressing and control needed for reading.



(a) FRONT END OUTPUT (SEQUENTIAL DATA)



(b) INPUT STEERING CONNECTIONS (ACCESSED RAMS)



(C) PAGE SELECTED (SEE LOAD MAP)

Fig. 25. Configuration 1 data streams, writing.

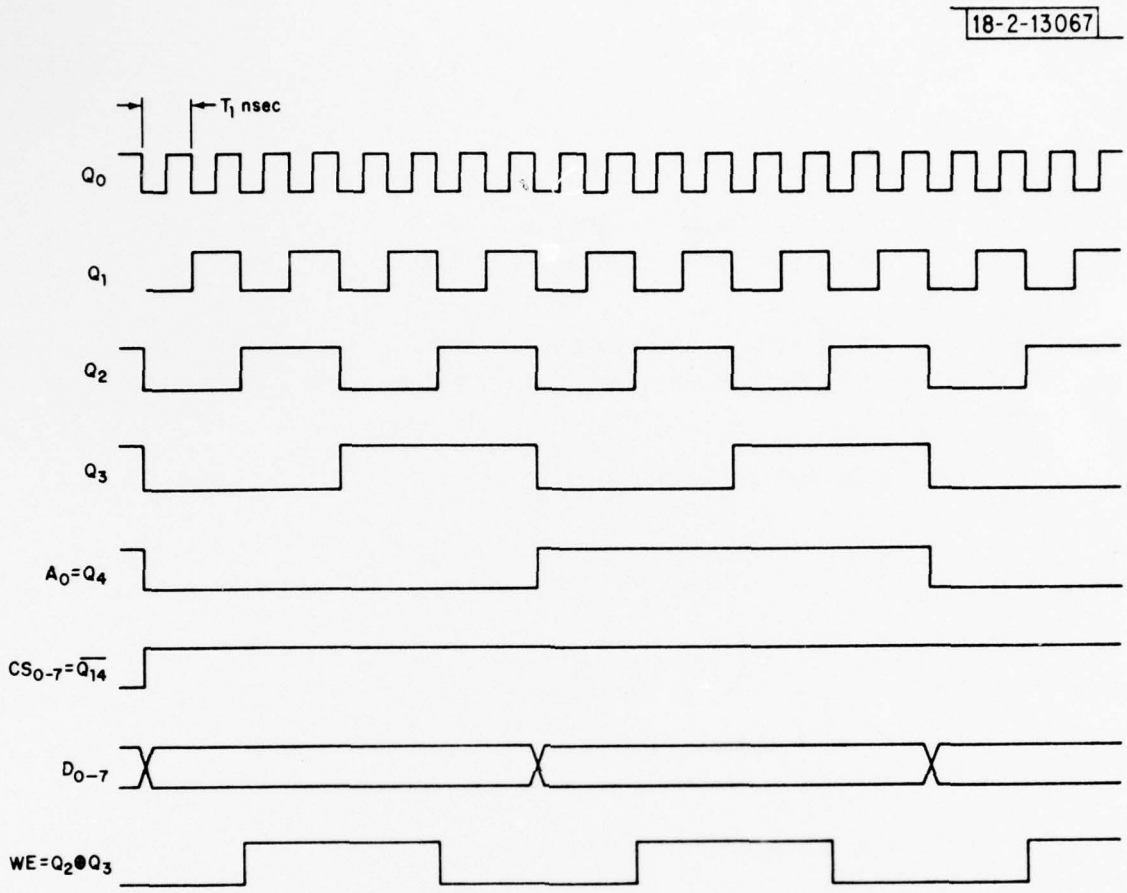


Fig. 26. Configuration 1 write timing diagram.

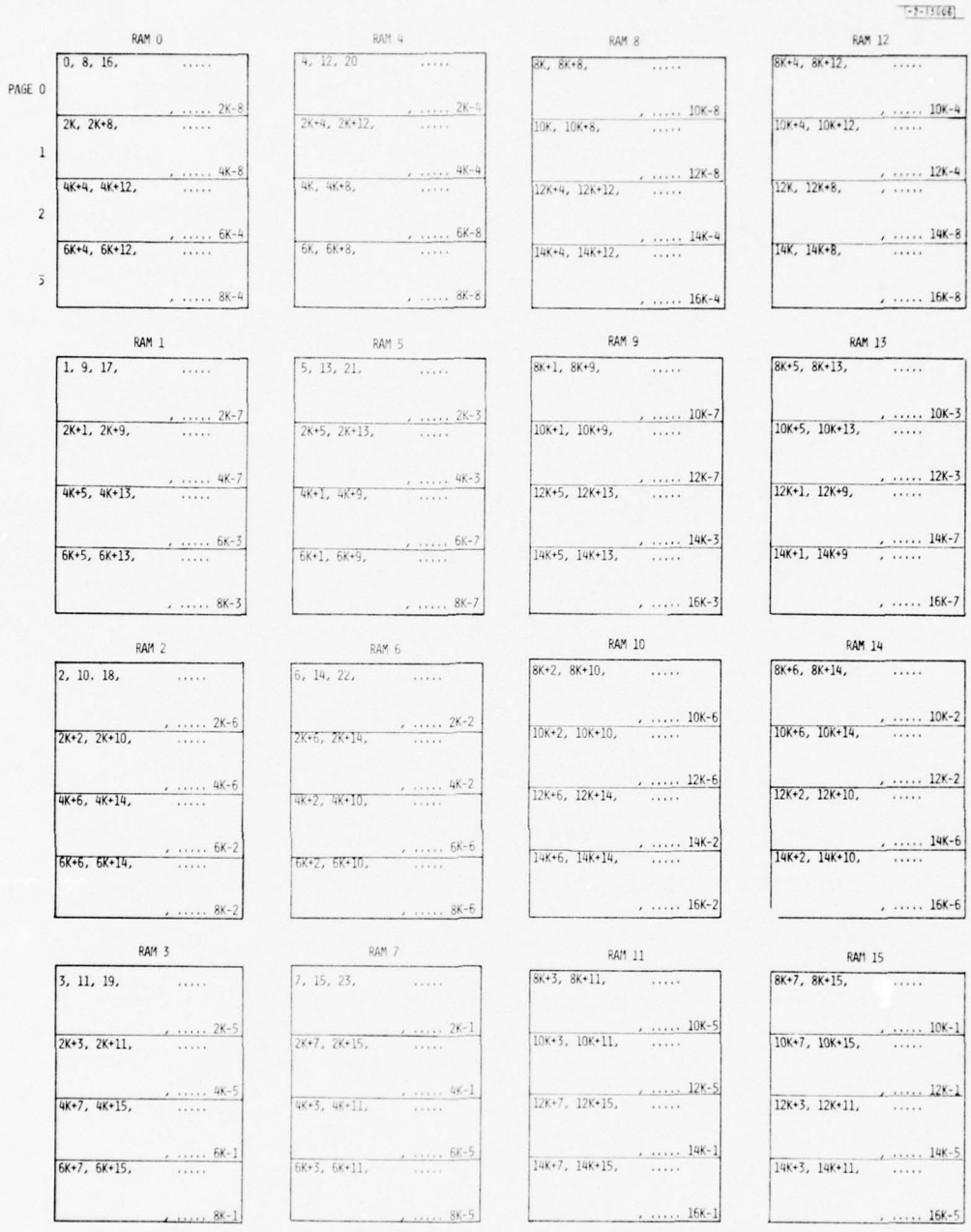


Fig. 27. Configuration 1 memory load map.

When reading the memory in configuration 1, the address/control counter is slowed by a factor of four and a delay register stack is introduced to permit address pipelining for improved speed. While page selection changes after every 2048 data points when writing, read operations require page switching on alternate RAM group accesses as shown in Figure 28(a). These same groups of 4 RAMS (the four RAM columns of Figure 27) receive the above-mentioned stack-delayed addresses and provide group access rates much higher than those possible using non-pipelined individual RAMS. Figure 28(b) indicates that while data points 0→3 are being read from RAM group 0→3, data points 4K→4K+3 are being read from RAM group 4→7, etc. The timing diagram of Figure 29 shows the relationship between the delayed addresses used within a given RAM group and the corresponding family of non-overlapping chip select pulses. The "data valid" window, while somewhat oversimplified, nevertheless conveys the idea that each RAM within a group contributes to the memory output 25% of the time. Strictly speaking, this window is narrowed due to the chip select access time requirement.

Figure 29 also indicates that the addressing within a page is altered at half the rate at which page selection changes. Consider, for example, the first two accesses of RAM 0: From Figures 4 and 27, it is evident that RAM 0 will supply data point 0 for rail D_0 when first accessed, and then will provide data point 4K+4 on output rail D_0' when accessed a second time. But data points 0 and 4K+4 are the first entries on pages 0 and 2 of RAM 0, hence the on-page address needn't change when page selection changes from 0 to 2.

18-2-13068

RAMS 0–3,8–11	0	2	0	•	•	•	•	2	1	3	1	•	•	•	•	3
RAMS 4–7,12–15	2	0	2	•	•	•	•	0	3	1	3	•	•	•	•	1

(a) PAGE SELECTED (SEE LOAD MAP)

T_{Q_2} → |nsec| ←

RAM 0–3	0	4K+4	8	•	•	•	•	6K-4	2K	6K+4	2K+8	•	•	•	•	8K-4
	↓	↓	↓					↓	↓	↓	↓					↓
	3	4K+7	11					6K-1	2K+3	6K+7	2K+11					8K-1
4–7	4K	4	4K+8					2K-4	6K	2K+4	6K+8					4K-4
	↓	↓	↓	•	•	•	•	↓	↓	↓	↓	•	•	•	•	↓
	4K+3	7	4K+11					2K-1	6K+3	2K+7	6K+11					4K-1
8–11	8K	12K+4	8K+8					14K-4	10K	14K+4	10K+8					16K-4
	↓	↓	↓	•	•	•	•	↓	↓	↓	↓	•	•	•	•	↓
	8K+3	12K+7	8K+11					14K-1	10K+3	14K+7	10K+11					16K-1
12–15	12K	8K+4	12K+8					10K-4	14K	10K+4	14K+8					12K-4
	↓	↓	↓	•	•	•	•	↓	↓	↓	↓	•	•	•	•	↓
	12K+3	8K+7	12K+11					10K-1	14K+3	10K+7	14K+11					12K-1

$T_{Q_{13}}$ nsec

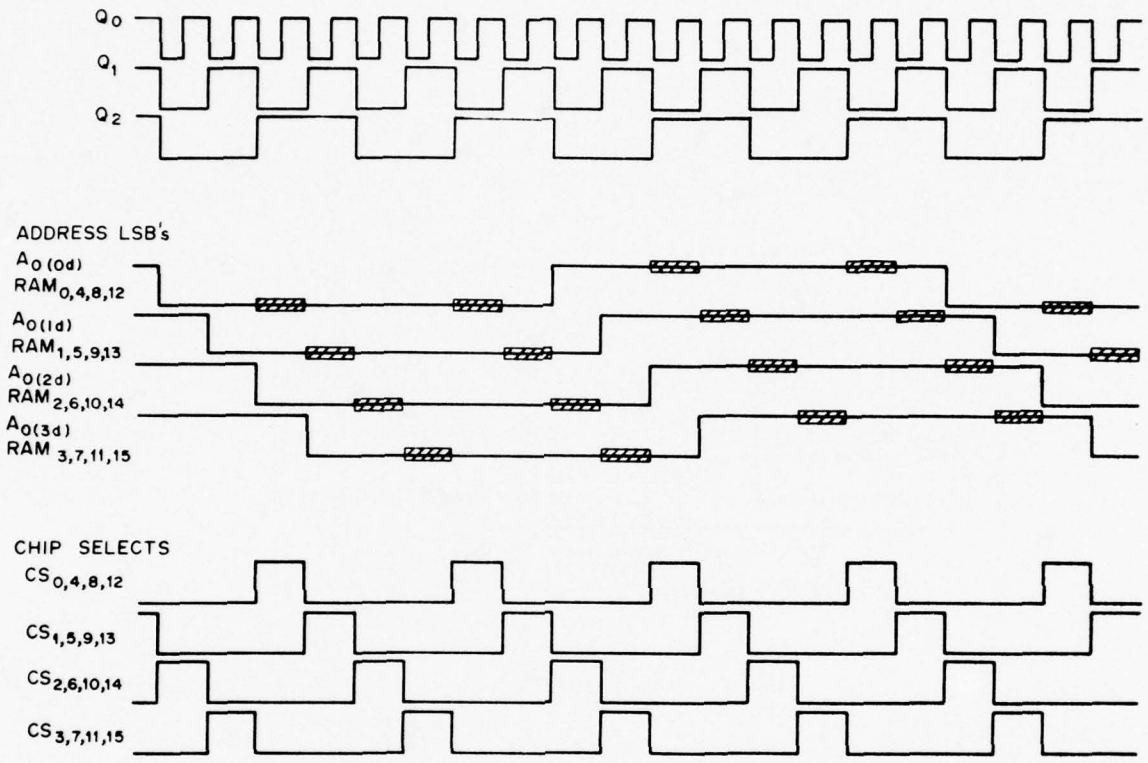
(b) RAM GROUP DATA OUT

DO	0	4	8	•	•	•	•	2K-4	2K	2K+4	2K+8	•	•	•	•	4K-4
	↓	↓	↓					↓	↓	↓	↓					↓
	3	7	11					2K-1	2K+3	2K+7	2K+11					4K-1
DO′	4K	4K+4	4K+8					6K-4	6K	6K+4	6K+8					8K-4
	↓	↓	↓	•	•	•	•	↓	↓	↓	↓	•	•	•	•	↓
	4K+3	4K+7	4K+11					6K-1	6K+3	6K+7	6K+11					8K-1
DO″	8K	8K+4	8K+8					10K-4	10K	10K+4	10K+8					12K-4
	↓	↓	↓	•	•	•	•	↓	↓	↓	↓	•	•	•	•	↓
	8K+3	8K+7	8K+11					10K-1	10K+3	10K+7	10K+11					12K-1
DO‴	12K	12K+4	12K+8					14K-4	14K	14K+4	14K+8					16K-4
	↓	↓	↓	•	•	•	•	↓	↓	↓	↓	•	•	•	•	↓
	12K+3	12K+7	12K+11					14K-1	14K+3	14K+7	14K+11					16K-1

(c) STEERED OUTPUT DATA

Fig. 28. Configuration 1 data streams, reading.

18-2-12898



RAM - GROUP OUTPUTS													
N_0	0	1	2	3	4K+4	4K+5	4K+6	4K+7	9	10	11	12	4K+12
N_1	4K	4K+1	4K+2	4K+3	4	5	6	7	4K+8	4K+9	4K+10	4K+11	13
N_2	8K	8K+1	8K+2	8K+3	12K+4	12K+5	12K+6	12K+7	8K+8	8K+9	8K+10	8K+11	12K+12
N_3	12K	12K+1	12K+2	12K+3	8K+4	8K+5	8K+6	8K+7	12K+8	12K+9	12K+10	12K+11	8K+12

Fig. 29. Configuration 1 read timing diagram.*
* the shaded regions of the address waveforms denote RAM "data valid" windows.

Figures 28(b) and (c) describe the result of output data steering. In configuration 1, the outputs of RAM groups 0→3 and 4→7, together with 8→11 and 12→15 are interchanged on alternate group accesses to form streams of contiguous data on output channel D_o through D_o'''.

Configurations 2,3

The differences between configurations 1 and 2 are summarized in Tables 2 and 3 and occur primarily in the areas of input steering, chip selection and page selection. The load map for configuration 2 is shown in Figure 30. Addressing and control has been defined to provide proper output formatting for two consecutive 8K transforms.

Configuration 3 requires the same high-speed front end performance as configurations 1 and 2 above, but differs in terms of input steering, chip selection and page selection. Unlike all other configurations, number 3 also utilizes an 8-page address organization (Figure 31); a consequence of requiring output data sets having as few as 4K points. If desired, three additional 4K sets may be stored with no modification of addressing and control. In such cases, the availability of status flags such as "1/4 full," "1/2 full," etc., is highly desirable.

Configurations 4,5,6

Configuration 4 is the first of three requiring two-channel inputs and a front end configured as a pair of 4:1 S/P converters. As noted in Table 3, the address/control counter is now scaled by a factor of 2 (double the speed of configurations 1→3) when reading. Incoming 8K data sets on channels DI and DI' are packed such that a single 16K transform may be performed (see Figure 4).

-2- 13007

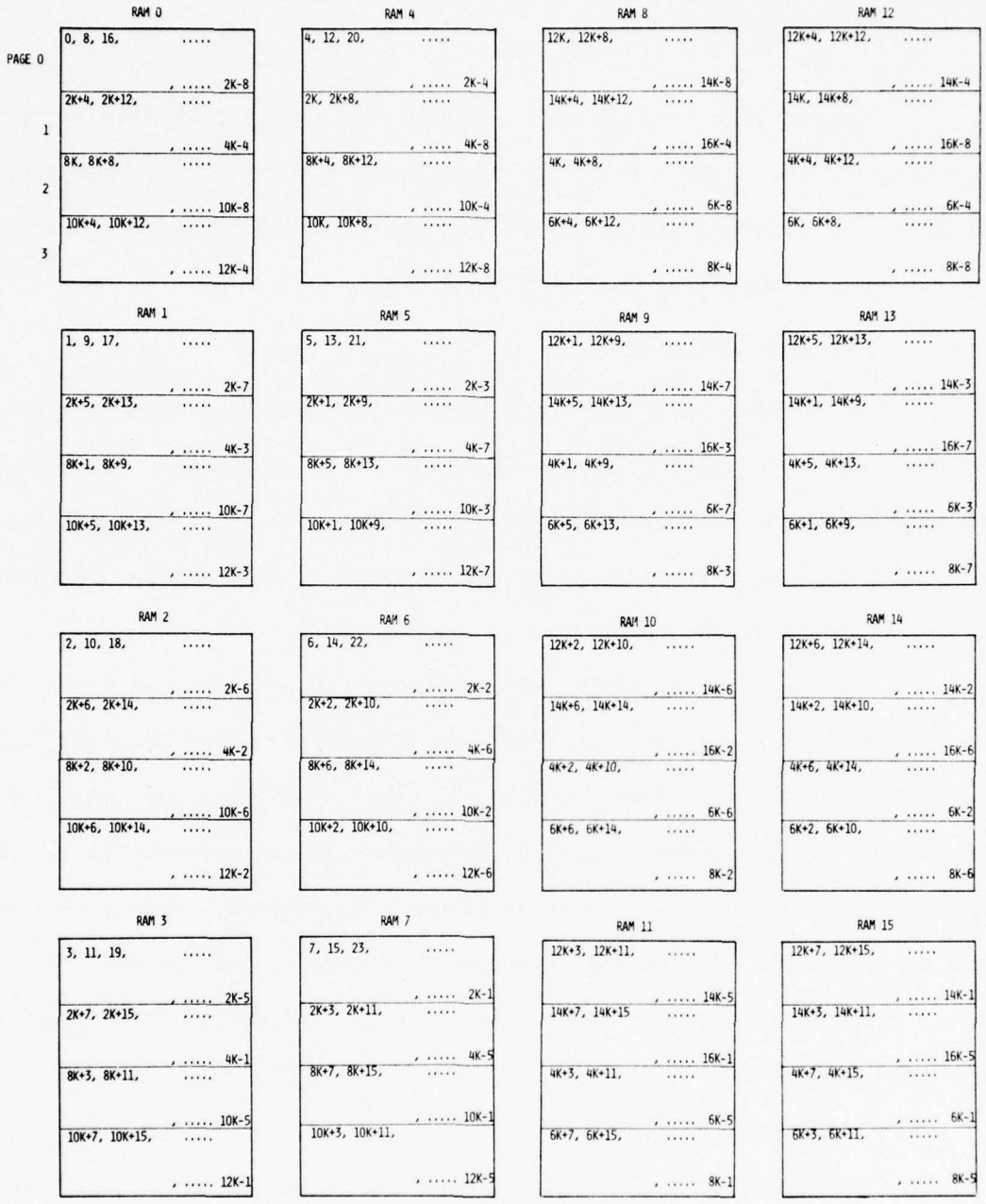


Fig. 30. Configuration 2 memory load map.

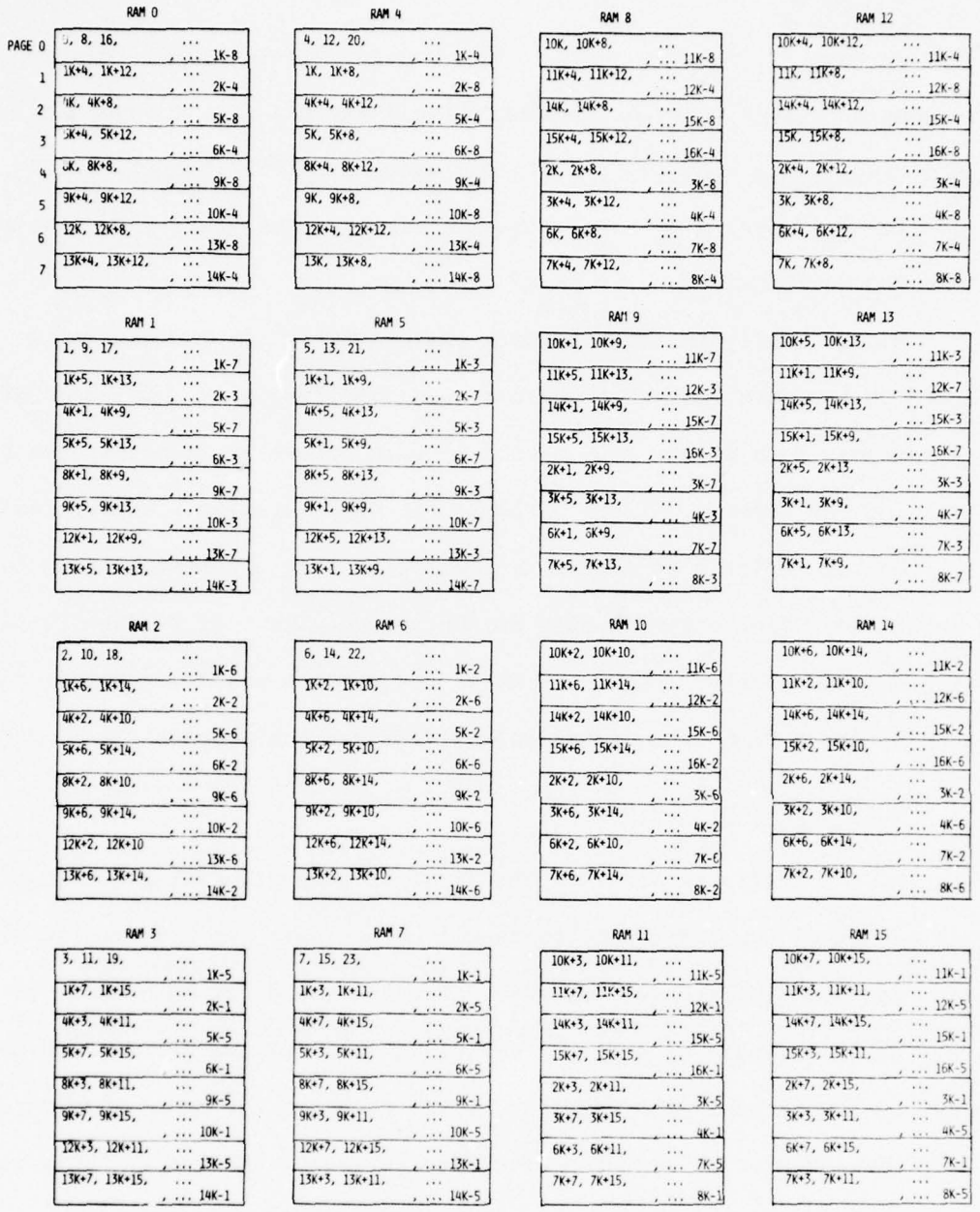


Fig. 31. Configuration 3 memory load map.

Unlike configurations 5 and 6, configuration 4 employs static input and output data steering, using RAMS 0→3 for storage associated with output rail D_o, RAMS 4→7 for D_o', etc. and only 8 RAMS are simultaneously written. The memory load map for configuration 4 is shown in Figure 32.

Module configuration 5 permits consecutive 8K transforms to be performed on two data sets which were stored simultaneously. Unlike configuration 4, dynamic data steering is needed to gain access to 2K-point data blocks when reading the buffer. Figure 33 shows the configuration 5 memory load map.

Configuration 6 operation is similar to that of configuration 5, except that input and output steering must switch at a higher rate to deposit 1K-point data segments in memory. Figure 34 shows how each 1K-point segment is distributed across 4 RAMS of a group (column) for eventual sequential read out.

Configurations 7,8,9

Module configuration 7 is the first of three remaining configurations wherein 4-channel input data and 4-channel output data are transferred at rates no less than the 30Ms/s convolver rate. The front end for this class of configurations is arranged as four 2:1 serial-to-parallel converters, and data is alternately deposited in the upper and lower banks of the memory array, unlike the left bank-right bank approach of configurations 1 through 6. This requirement arises because data associated with all four output rails must be located such that the 4-RAM group pipelined-read approach used in all other configurations may be retained. As indicated earlier, we trade some additional input circuit complexity for output circuit simplicity. That complexity resides within the

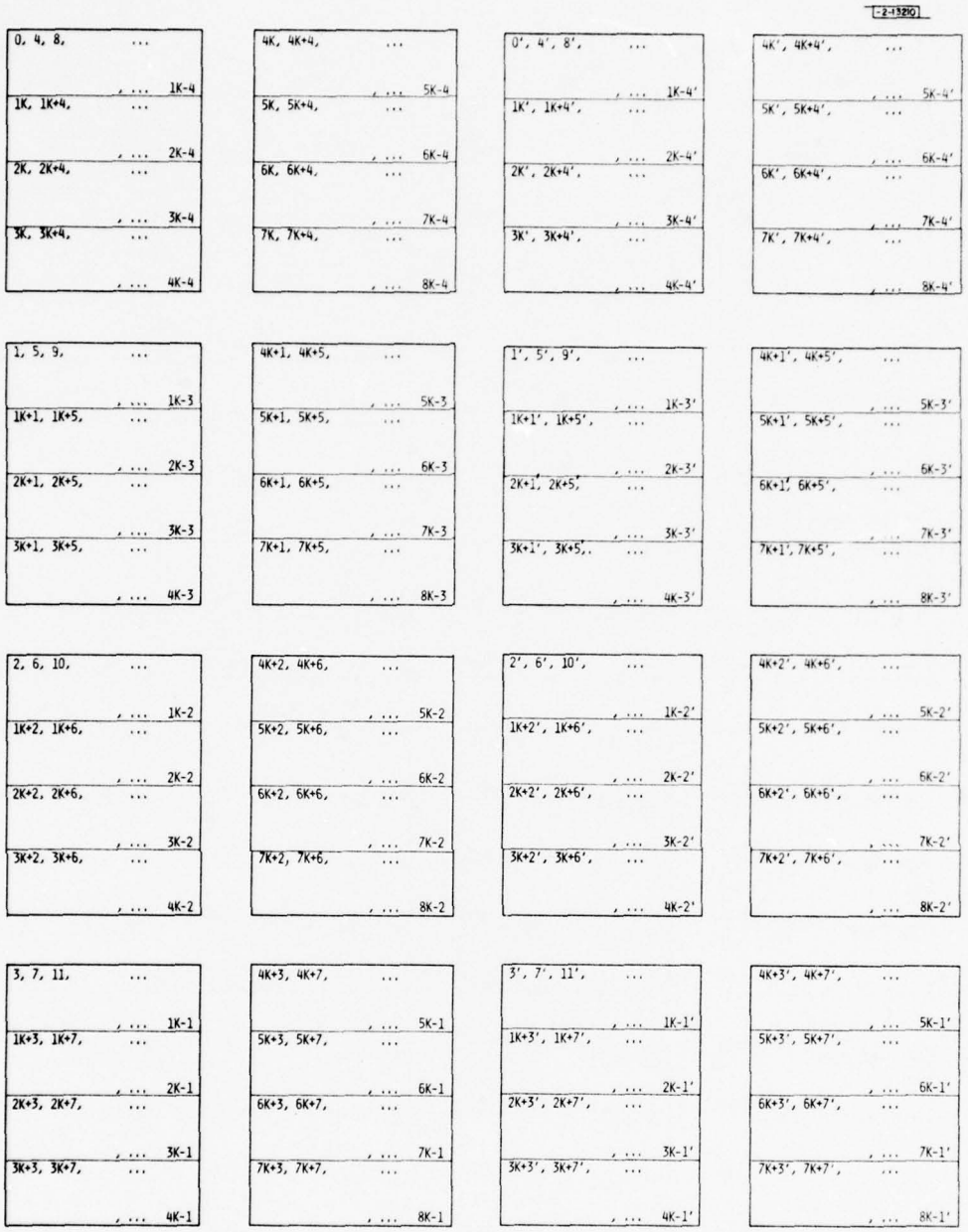


Fig. 32. Configuration 4 memory load map.

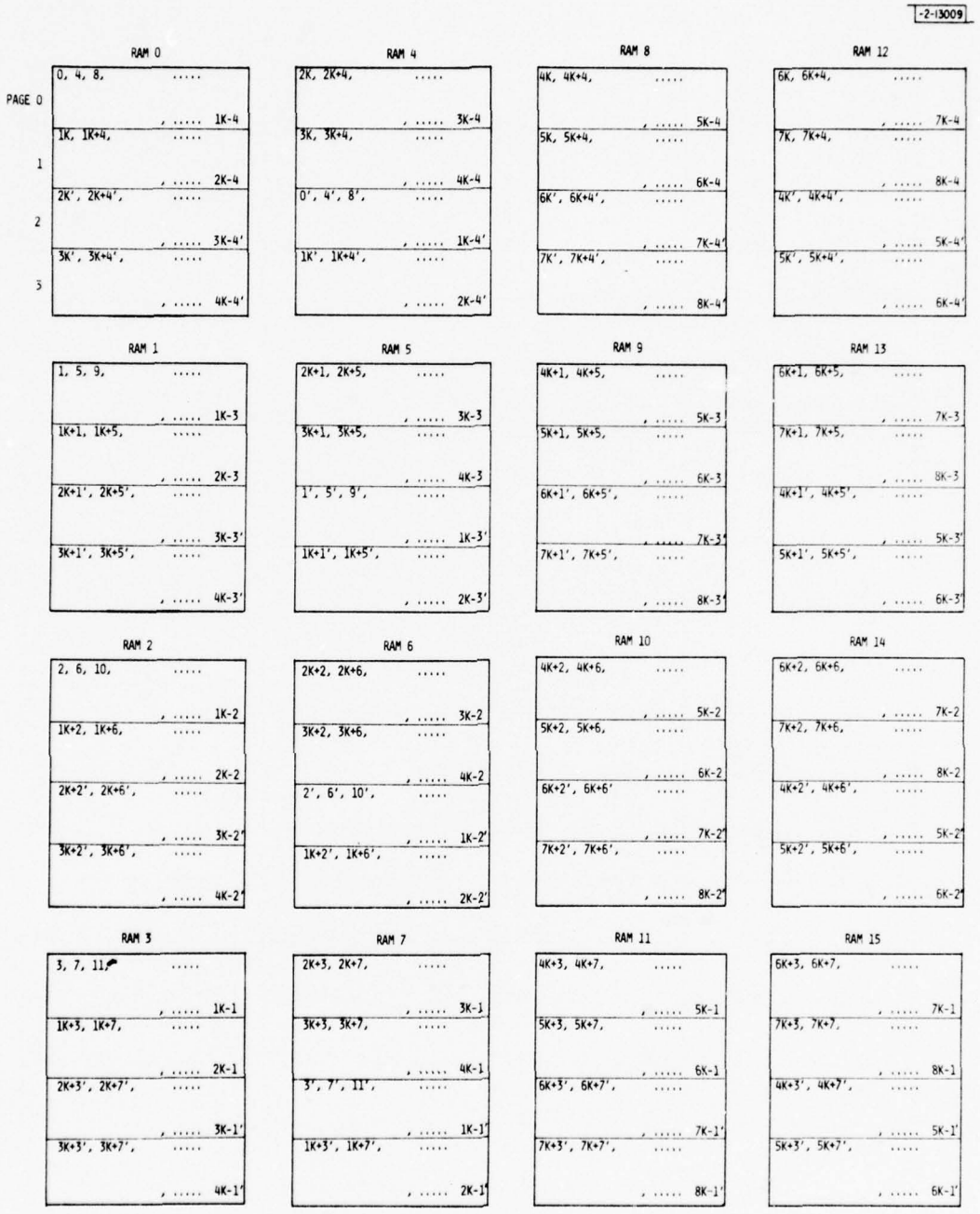


Fig. 33. Configuration 5 memory load map.

-2-13009

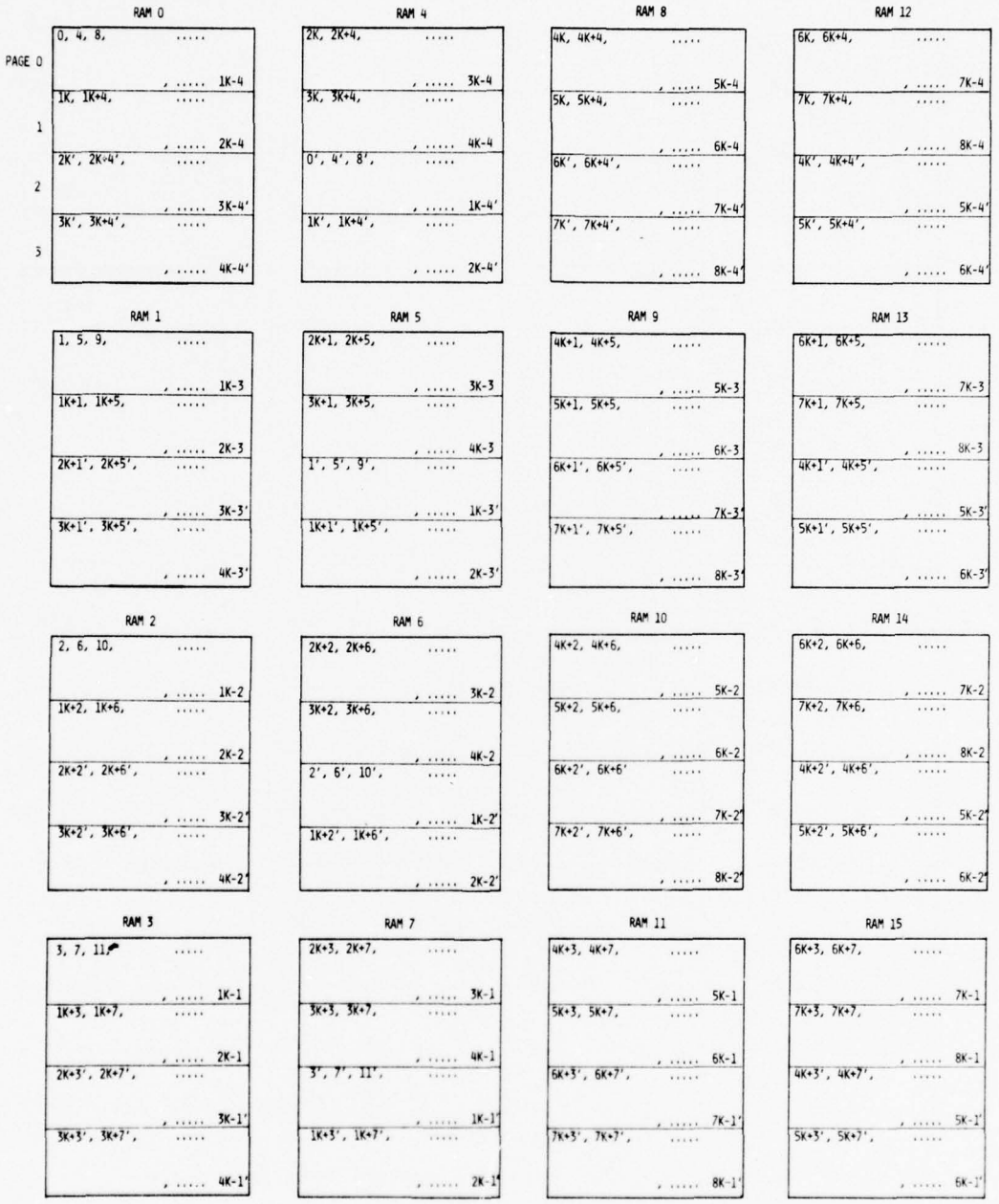


Fig. 33. Configuration 5 memory load map.

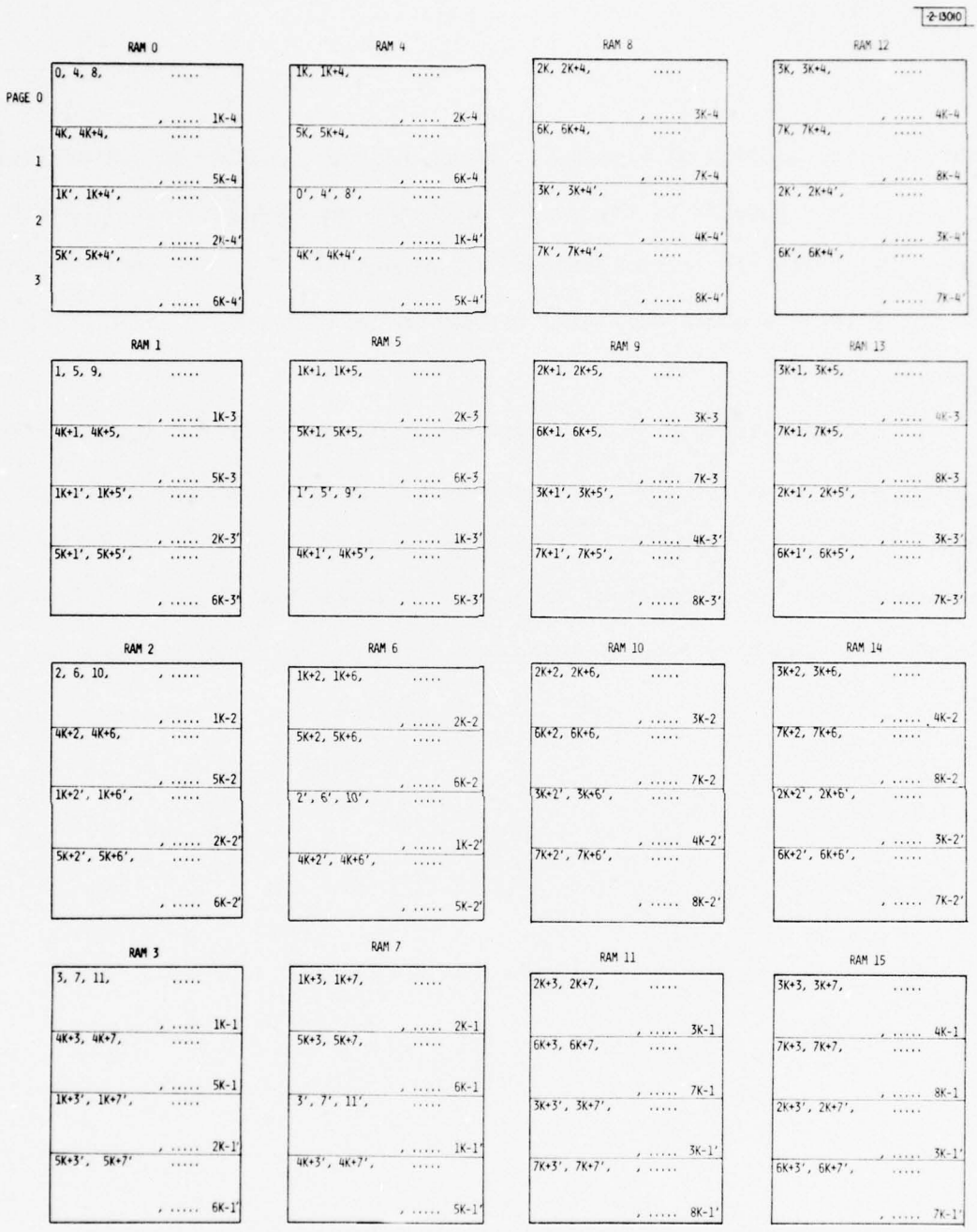


Fig. 34. Configuration 6 memory load map.

input steering section of Figure 21, which otherwise would use 2:1 multiplexers exclusively. Figure 35 is the memory load map for configuration 7, and Tables 2 and 3 delineate the control states. Configuration 7 is the only one in which incoming data on a given rail will ultimately be read out on the corresponding output rail.

Configurations 8 and 9 permit an individual buffer module to accept 4K data sets on each of four input channels, and to format the data upon readout to permit either two 8K transforms or four 4K transforms, respectively. These configurations store data in exactly the same way (Figure 36) and differ only slightly in their control states for reading, as shown in Table 3.

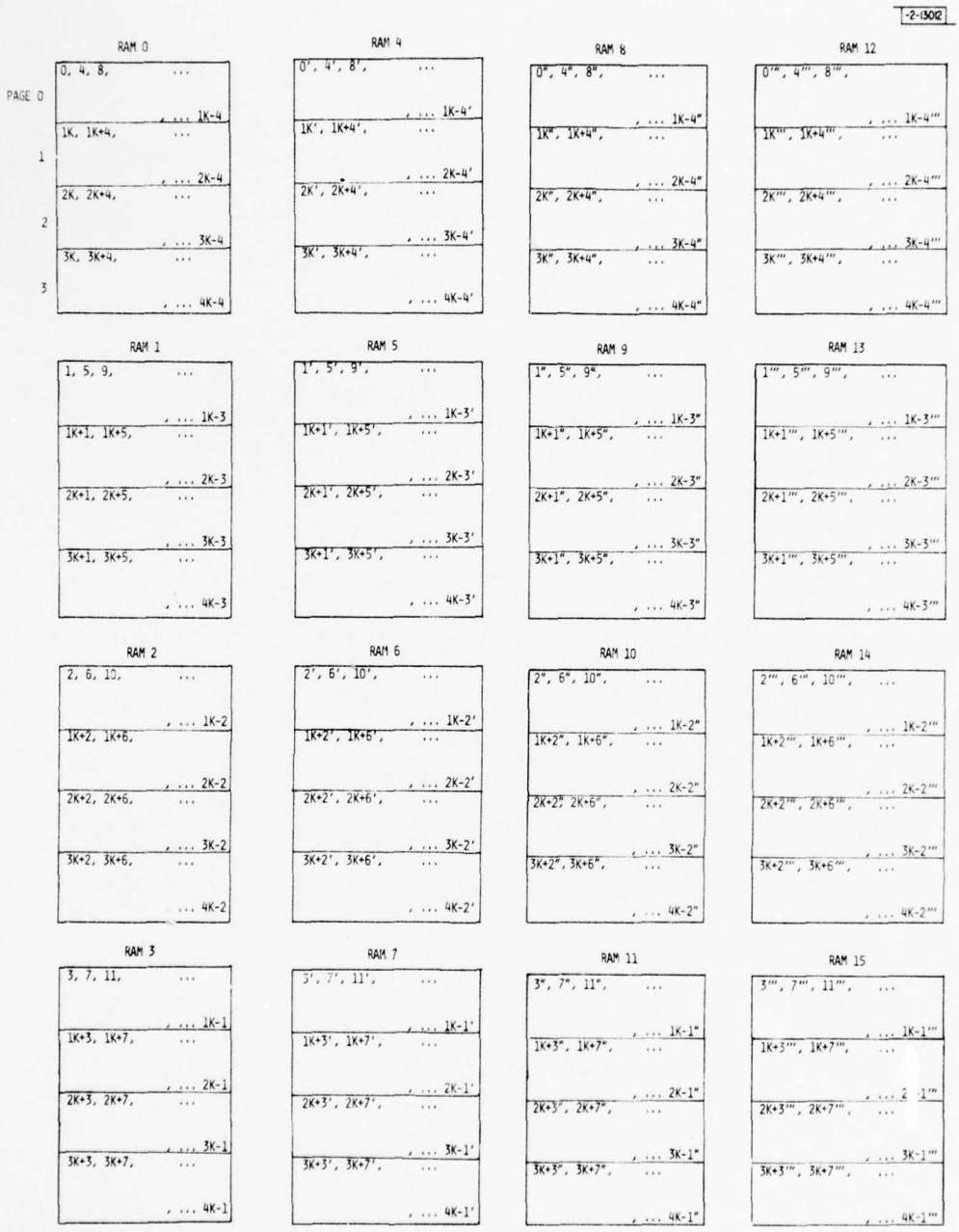


Fig. 35. Configuration 7 memory load map.

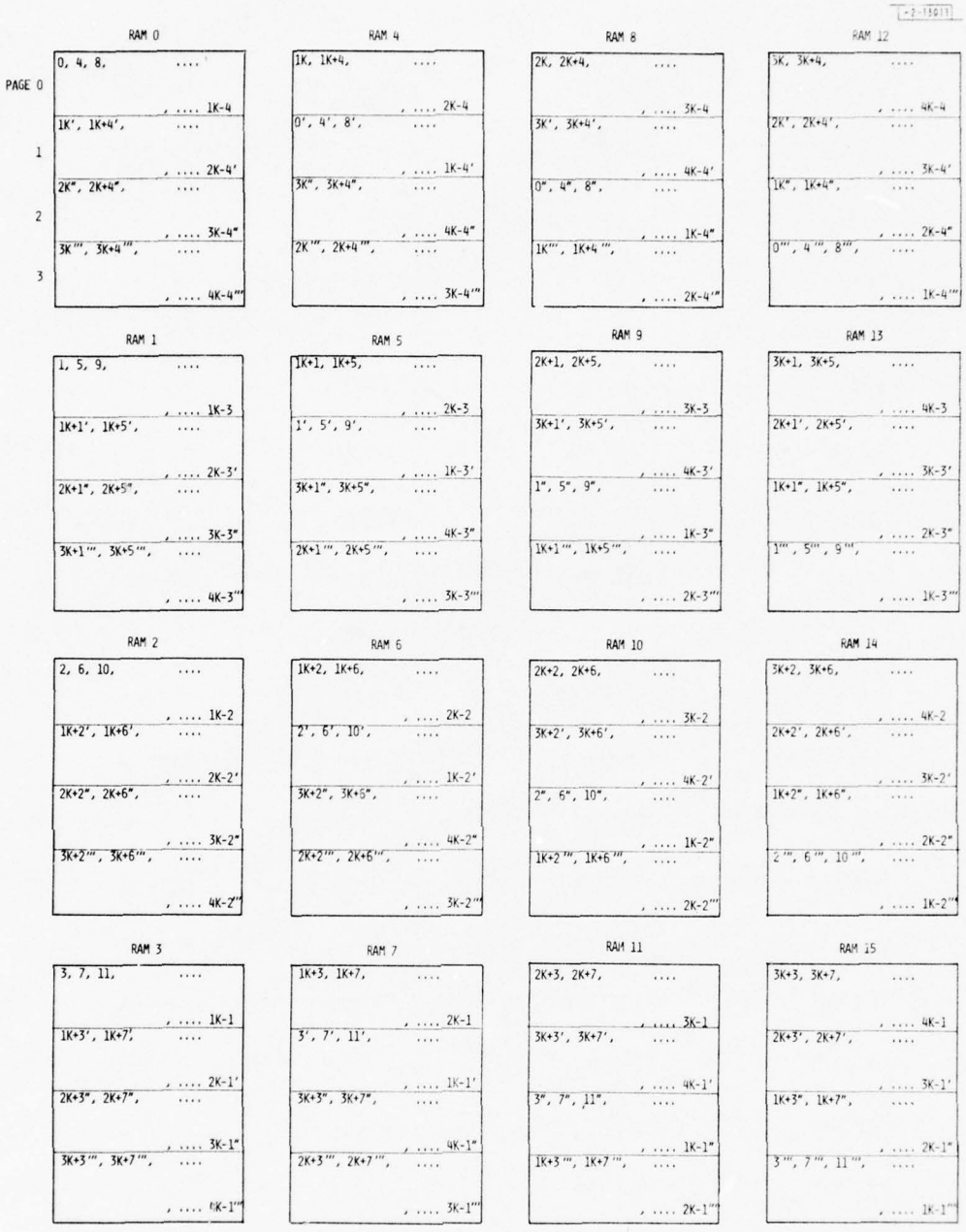


Fig. 36. Memory load map, configurations 8, 9.

V. HARDWARE

Having outlined the essentials of the basic buffer module, it remains to determine the performance limits (principally speed) using available, off-the-shelf components. The following discussion on hardware examines three high performance subsections of the buffer module.

Worst-case analysis and experimental results are included for these selected portions assuming an all-ECL memory realization. The hardware discussion concentrates on those subsystems within which there exists most technological risk (speed uncertainty); namely, the front end, memory array and output steering.

A. Front End

This section describes a baseline ECL 10K front end whose input speed can keep pace with the A/D converters to be used in the signal processor system. An alternative ECL 100K design permitting higher IB input speeds is also discussed.

The ECL 10K front end logic diagram is shown in Figure 37. Unlike the conceptual diagram of Figure 17, we now include specific devices by generic part designation. The front end is a programmable serial-to-parallel converter, followed by a parallel holding register, which accepts differential inputs from the A/D converters associated with up to four radar channels. The ECL 10000 4-bit shift register forms the basis for the design. For either one or two-channel modes, each 4-bit register accepts serial inputs via 10173 multiplexers, but when four channels are active, the parallel entry feature is used.

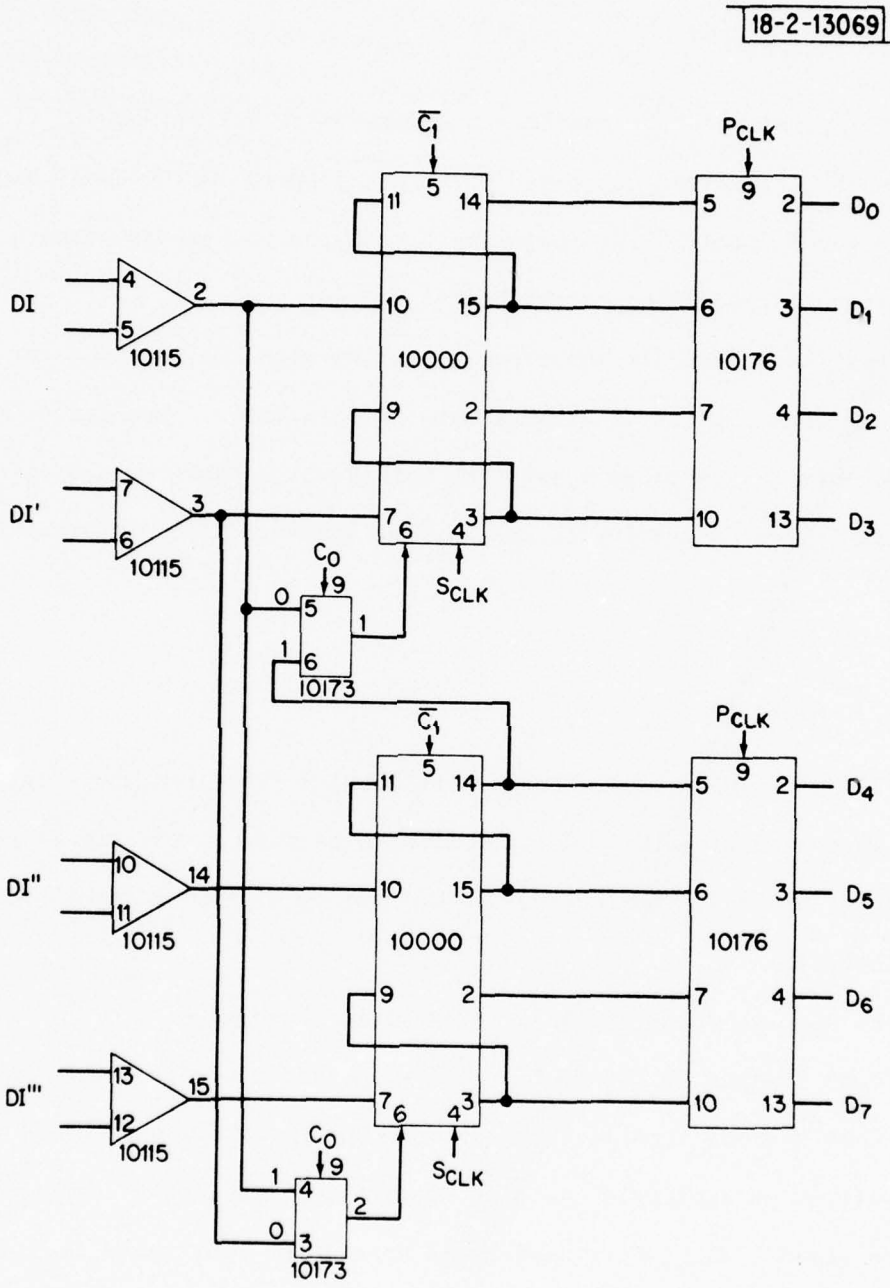


Fig. 37. ECL 10K front end logic diagram.

Potential front end speed is directly related to the timing requirements of the 10000 and 10176 registers. In the steady state, the maximum data transfer rate is a function of register set-up and hold times, plus allowances for signal degradation along with clock and control skews. Signal degradation refers to less-than-ideal (but acceptable) waveform quality such as attenuation or slowed edge speeds resulting from operation at high rates with moderate fan-outs. A clock skew allowance is made in recognition of the uncertainty of precise clock arrival times throughout a system of moderate size.

The calculation of expected speed in logic systems is strongly influenced by the level of designer conservatism conditioned by prior experience. In the timing budget tallied below, front end speed capability is conservatively rated while meeting minimum performance requirements with high confidence.

TABLE 4

DELAY TOTALS FOR SPEED-CONSTRAINING FRONT END PATHS

10000 timing		10176 timing		Definitions
T_{SU} =	1.5 ns	T_{SU} =	2.5 ns	T_{SU} = register input data set-up time prior to clock
T_{SKEW} =	2.5	T_{SKEW} =	2.5	T_{SKEW} = time allowance for skew
T_{DEG} =	2.0	T_{DEG} =	2.0	T_{DEG} = time allowance for waveform degradation
T_{H} =	1.0	T_{H} =	1.0	T_{H} = register input data hold time after clock
	7.0 ns		8.5 ns	

From Table 4, 10176 set-up and hold times limit the expected front end data rate to ~117 Ms/s.

A 1-bit microstrip-interconnected front end has been built and tested. Several combinations of 10000 and 10176 devices were operated while maintaining the timing budgets of Table 4. Observed input data rates were not less than 125 Ms/s.

It should be noted that the 10176 could certainly be replaced by the faster but less dense 10000 part, which would increase expected front end speed to 142 Ms/s. However, additional device and packaging expense would be incurred.

The IB can have a single-channel input rate in excess of 142 MHz by using an ECL 100K front end such as the one shown in Figure 38. Using delay budget assumptions similar to those of Table 4 but for 100141 devices, the ECL 100K front end should accommodate input rates in the 250-300 Ms/s range. These rates exceed the capability of the 10K memory array described in Section IV.

B. Memory Array

This design assumes the use of the ECL 10415A 1024 by 1-bit IC. When reading the buffer, the 10415A's worst-case chip select access time of 10 ns is of particular significance. Timing relationships for writing into the buffer are shown in Figure 39. In the steady state, memory write cycle time is determined principally by address set-up time, write pulse width, chip select hold time, and an allowance for signal condition and skew. Budgeting

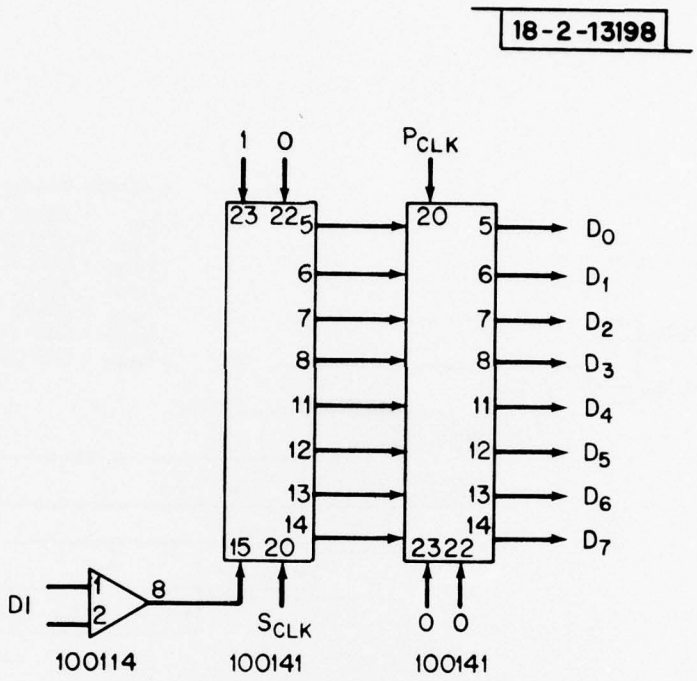


Fig. 38. ECL 100K front end logic diagram.

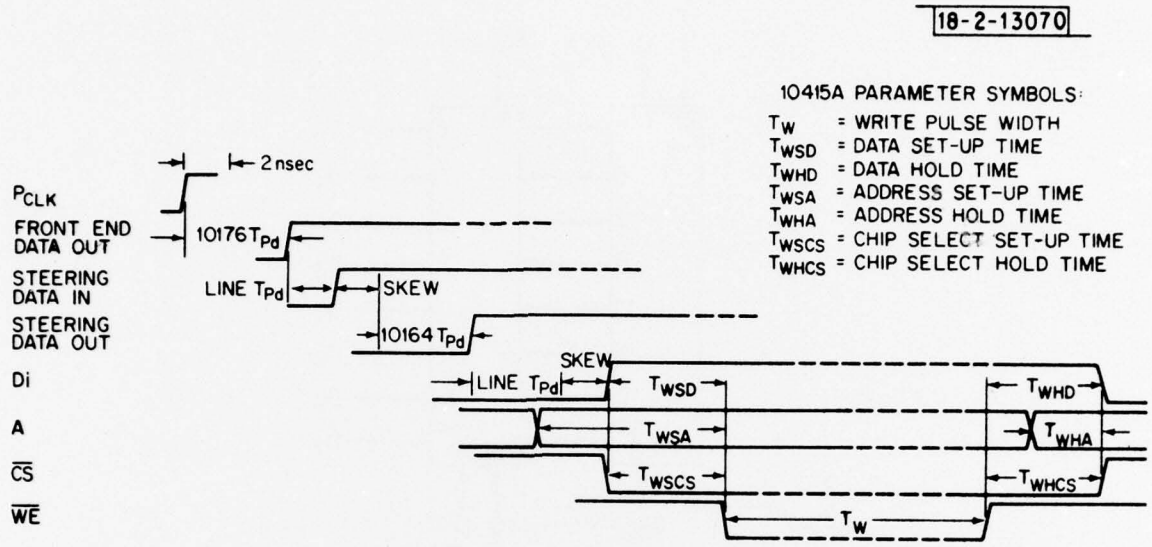


Fig. 39. ECL 10415A RAM write timing diagram.

4 ns for the latter and using worst-case 10415A parameters, the basic memory write cycle time becomes 42 ns, which translates into maximum attainable one, two and four-channel input rates of 190, 95 and 47 Ms/s, respectively.

Figure 40 compares the memory array writing speed as a function of the number of input channels with the performance obtainable using any of three front ends. Points 1 and 2 represent the 4-channel and 2-channel input cases, where speed is constrained by memory, and hence any front end will suffice. For single channel operation, the ECL 10K front ends limit input rates at 117 and 142 Ms/s, whereas at 190 Ms/s, the 100K front end would be required. In this case, the performance limit would be determined by the memory array.

C. Output Steering

Pipelined addressing is employed on 4-RAM groups to achieve buffer read speeds higher than those possible using conventional methods. Figure 41 depicts a 4-RAM group featuring 4-phase addressing, non-overlapping chip selects and wire-OR'd outputs. Basically, given the sequential nature of stored data, we take advantage of the relatively fast 10415A chip select access time (compared to address access) and commutate from RAM A to RAM D. The 4:1 data multiplexers of Figure 23 have been supplanted by 9:1 multiplexers, as would be needed in System 2. The somewhat irregular 9:1 mux can be built using a 10164 8:1 multiplexer with a simple AND gate, and its delays are those of the 10164. Figure 42 shows the relationship between RAM A and RAM B signals, which also exists between RAMS B and C, C and D, and D and A.

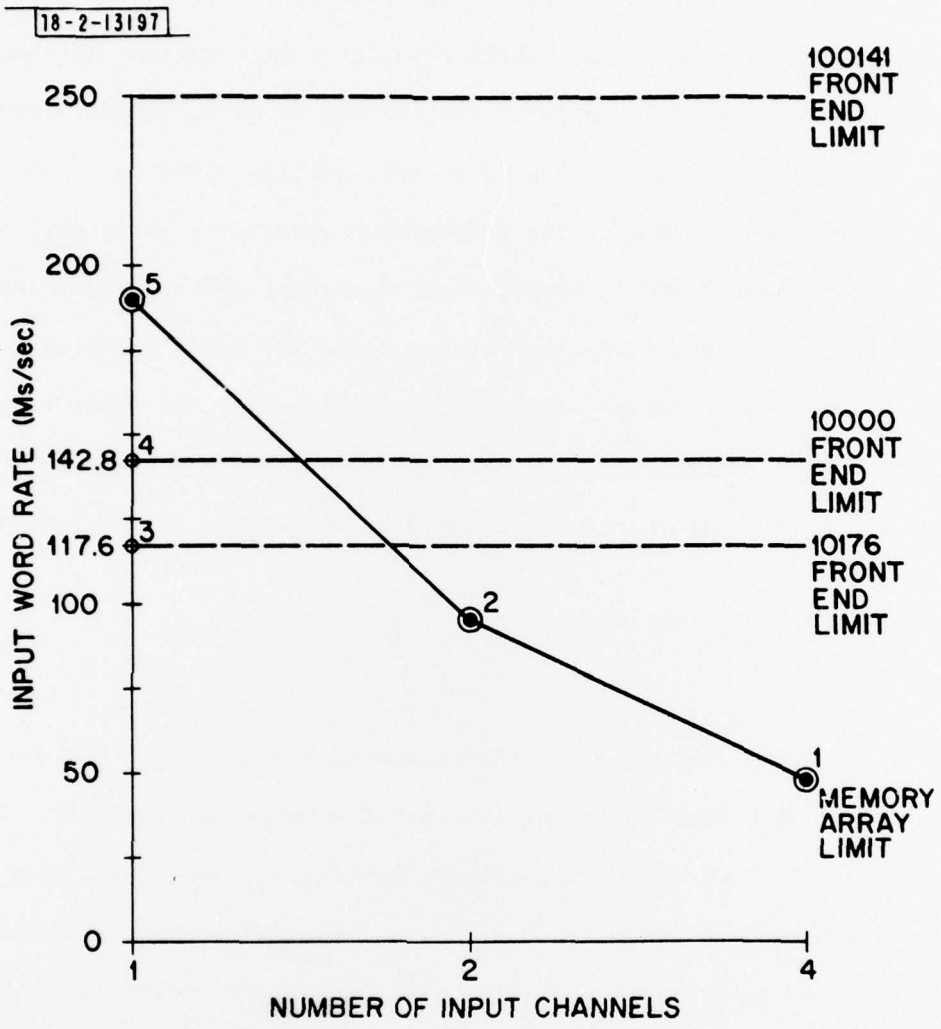


Fig. 40. Buffer module input rates as a function of front end speed, memory array speed, and number of channels.

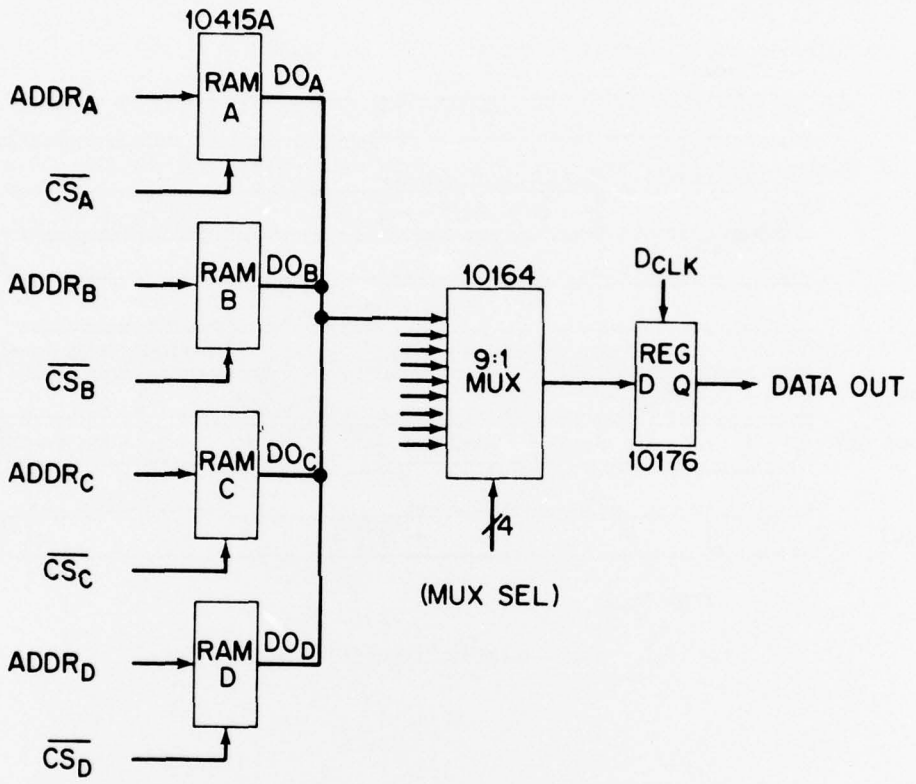


Fig. 41. Fast memory read logic diagram.

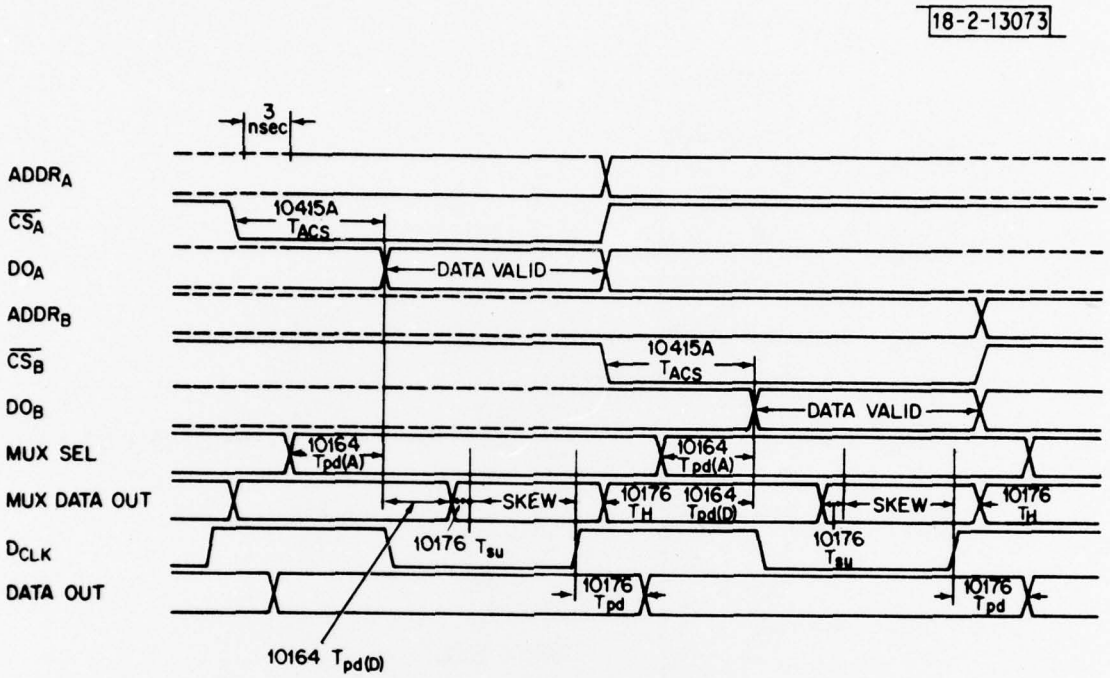


Fig. 42. Fast memory read timing diagram.

The delay budget for determining worst-case buffer access time is given in Table 5 below, and includes the usual device and interconnection delay allowances. An additional 1 ns penalty is assessed for increased propagation delay when wire-ORing RAM outputs. A relatively narrow time window exists for which data appearing at the 10164 output is valid and may be registered by the 10176. Based on worst-case calculations, the expected buffer memory read rate is 42.5 Ms/s.

TABLE 5

DELAY TOTALS FOR SPEED-CONSTRAINING OUTPUT STEERING PATH*

10415A	T_{ACS}	10 ns
	$T_{WIRE-OR}$	1
10164	T_{PD}	4.5
10176	T_{SU}	2.5
	T_{SKEW}	2
	T_{PATH}	2
10176	T_H	1.5
		23.5 ns

An experimental 1-bit slice of memory and output steering logic has been built and tested. Maintaining worst-case timing margins wherever possible, achievable data rates were typically 45 Ms/s.

*10415A $TAA_{min} = TRCS_{min} = 0$ assumed

A preliminary sizing of System 2 (the 9-module system) has been done to gain some notion of total circuit count, the physical volume, and power dissipation. The results are summarized in Table 6.

TABLE 6

PRELIMINARY SIZING OF SYSTEM 2

ITEM	# IC's	#CKT CARDS	POWER (WATTS AC)	VOLUME (FT^3)
1 module	1009	10	565	5.85
9 modules	9081	90	5085	52.65
parity, selectors	646	7	275	3.7
IB totals	9727	97	5360	56.4

Assumptions:

- standard backplane/circuit card packaging
- forced air cooling
- high efficiency switching power supplies
- 85in^2 printed circuit cards
- 10in^3/IC for total cabinetry*

*This figure includes power supplies and air ducting, and is consistent with DCS packaging density.

The expected replication of like kinds suggests the use of printed circuit cards for all but one-of-a-kind control cards, which should be wire-wrapped. In anticipation of a high number of address/control lines throughout the system, all backplanes are likely to be of the multilayer printed circuit type.

VI. SUMMARY AND CONCLUSIONS

The Input Buffer (IB) module designs described in this report provide the first level buffering for the Signal Processor discussed in Section I. A basic buffer module is defined which has a selectable number of input channels and data storage formats. A module delivers four parallel outputs to the DCS.

Using the basic building block in its single input configuration, three buffer memory designs are presented. The first uses double-buffered modules on each of four input channels. The second has triple-buffered modules on each of three input channels. The third system is a variation of the second system which allows the multiple, overlapped data collection needed in all range processing.

Prototype hardware experiments have been conducted to determine the speed limits within the module front end, memory array and output data steering sections. Results indicate that a front end based on the device type 10176 holding register can support input rates of 125 Ms/s per channel. A second ECL 10K design using the device type 10000 holding register (less integration and hence a larger system) should operate at 142 Ms/s. An ECL 100K front end using device type 100141 should permit input speeds in excess of 250 Ms/s. However, experimental results show that the ECL 10K memory array, given that the front end provides 8:1 data rate reduction (8 of 16 RAMS/bit are written simultaneously), limits at 190 Ms/s. Therefore, the memory array constrains input speed at 190 Ms/s when 8:1 demultiplexing is employed.

Pipelined addressing on 4-RAM groups within the memory array permits buffer output rates greater than those using conventional addressing. Experimental results indicate typical output rates of 45 Ms/s per DCS rail.

Lastly, it must be emphasized that the designs presented here include many options and many alternative configurations. Any actual IB would select only a subset of the design alternatives discussed in this report.

ACKNOWLEDGEMENTS

The author is grateful to J.M. Frankovich and A.H. Anderson for their valuable insights and suggestions during the development of the input buffer designs presented here. Particular thanks are extended to R.J. Purdy who reviewed much of the manuscript and offered helpful criticism. E.J. Aho and S.R. Tringale made significant contributions in the area of prototype hardware experiments, and their work is appreciated.

REFERENCES

1. R.J. Purdy et al., "Digital Signal Processor Designs for Radar Applications," Technical Note 1974-58, Vols. 1 and 2 Lincoln Laboratory, M.I.T. (31 December 1974), DDC AD-B001419-L (Vol. 1) and AD-B001420-L (Vol. 2).

2. The ECL Handbook (Fairchild Semiconductor, July 1974).

3. Semiconductor Data Library, Vol. 4 (Motorola, Inc., 1974).

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ESD-TR-77-26	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* Input Buffer Designs for a Radar Signal Processor		5. TYPE OF REPORT & PERIOD COVERED Technical Note 6. PERFORMING ORG. REPORT NUMBER Technical Note 1977-3
7. AUTHOR(s) Albert H. Huntoon	TN-1977-3	8. CONTRACT OR GRANT NUMBER(s) F19628-76-C-0002
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lincoln Laboratory, M.I.T. P.O. Box 73 Lexington, MA 02173		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project No. 8X363304D215
11. CONTROLLING OFFICE NAME AND ADDRESS Ballistic Missile Defense Program Office Department of the Army 1320 Wilson Boulevard Arlington, VA 22209	87 p.	12. REPORT DATE 20 Jan 1977 13. NUMBER OF PAGES 88
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)* Electronic Systems Division Hanscom AFB Bedford, MA 01731		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

input buffer
radar signal processor
digital processing techniques

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Three Input Buffer designs which provide intermediate storage between the Analog to Digital Converters and the Digital Matched Filter of a Radar Signal Processor are presented. All designs use a basic all-ECL buffer module. Prototype hardware experiments indicate that with 8:1 input data multiplexing, ECL 10K technology will yield input rates up to 142 Ms/s per channel, and output rates of 45 Ms/s per convolver rail.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

207 650